



# CSCD43: Database Systems Technology

## Lecture 8

*Wael Aboulsaadat*

Acknowledgment: these slides are based on Prof. Garcia-Molina & Prof. Ullman slides accompanying the course's textbook.

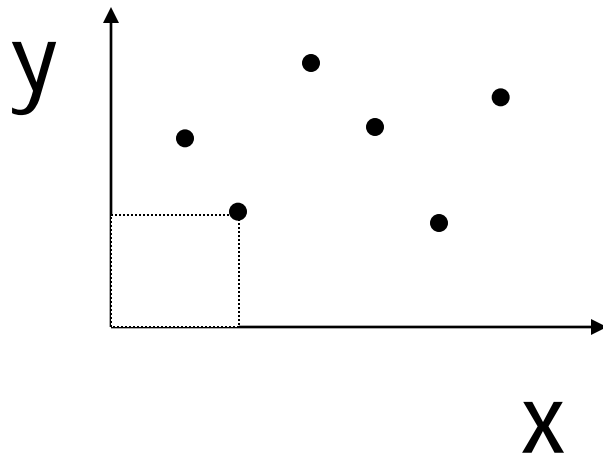


## Topics

- Conventional Indexes
- B-trees
- Hashing Schemes
- Multidimensional Indexes

## Interesting application:

- Geographic Data



DATA:

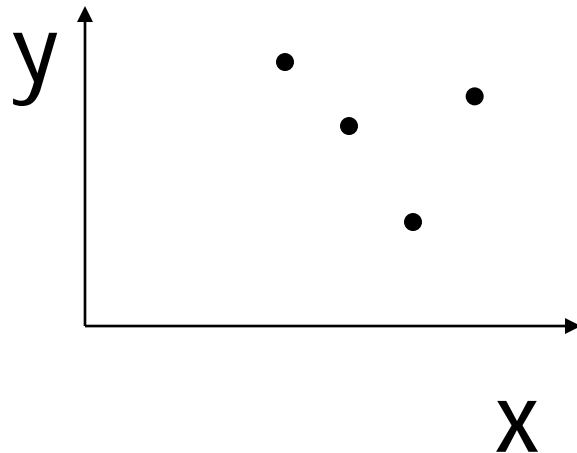
$\langle X_1, Y_1, \text{Attributes} \rangle$

$\langle X_2, Y_2, \text{Attributes} \rangle$

•  
•  
•

## Queries:

- What city is at  $\langle X_i, Y_i \rangle$ ?
- What is within 5 miles from  $\langle X_i, Y_i \rangle$ ?
- Which is closest point to  $\langle X_i, Y_i \rangle$ ?



# Multi-key Index

Motivation: Find records where

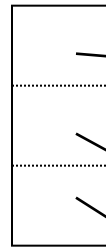
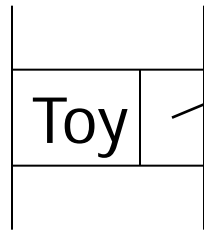
DEPT = "Toy" AND SAL > 50k

OR

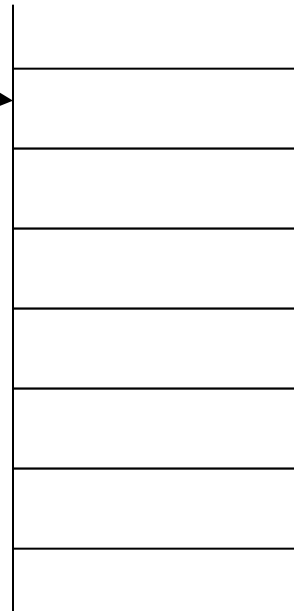
CREATE INDEX foo ON R(A,B,C)

Query: Get employees in  
(Toy Dept)  $\wedge$  (2nd floor)

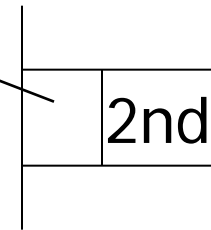
Dept. index



EMP



Floor index

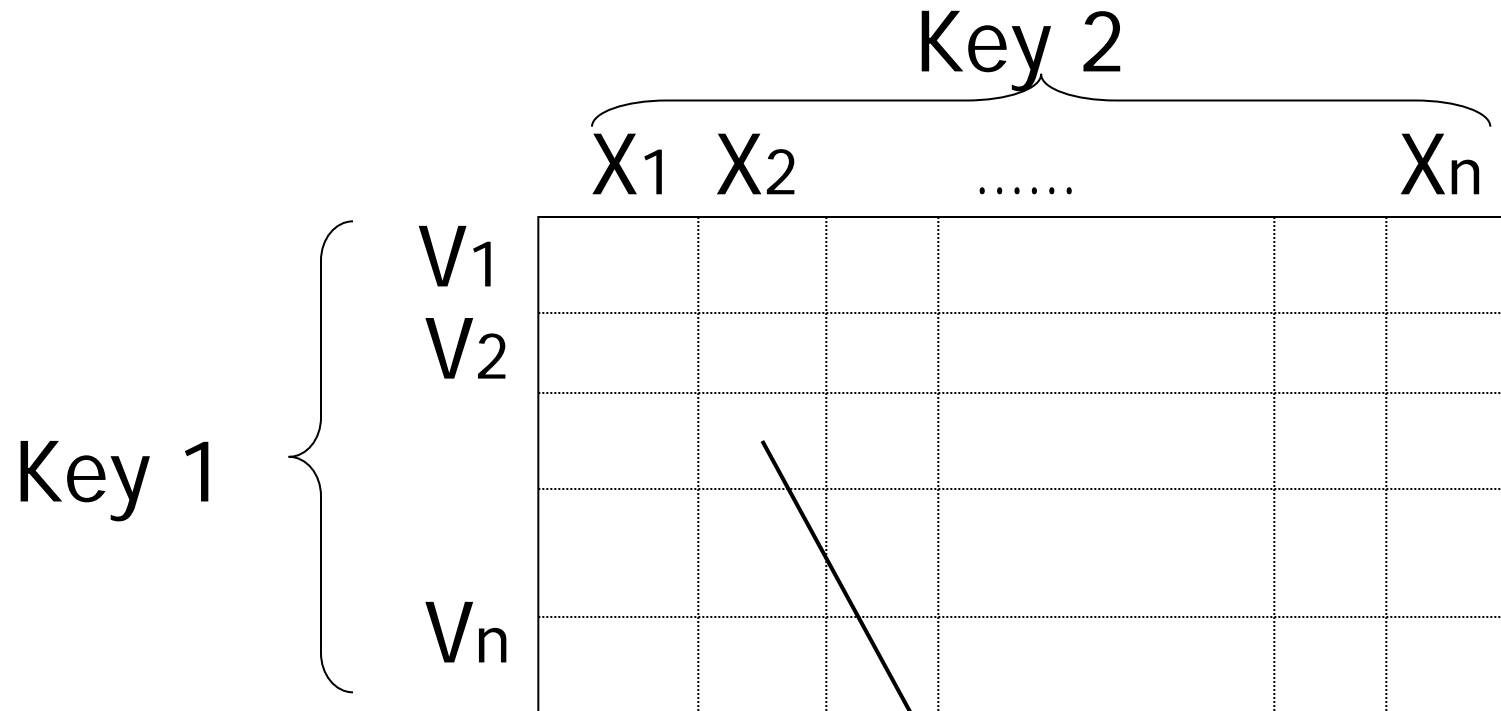




# Types of multi key indexes

- Grid
- Partitioned hash
- MultiKey Trees

# 1) Grid Index



To records with key1 =  $V_3$ , key2 =  $X_2$

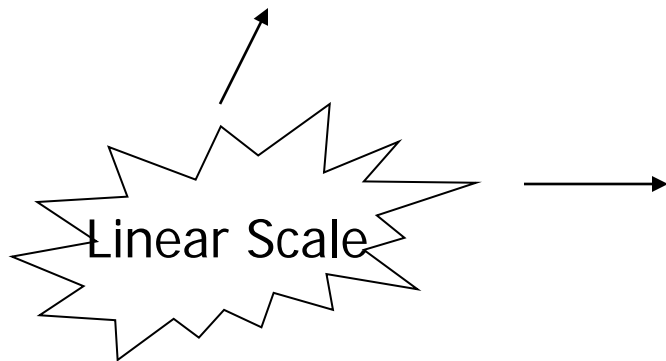
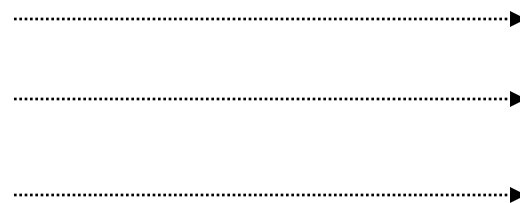


# Index grid on Values or Value Ranges

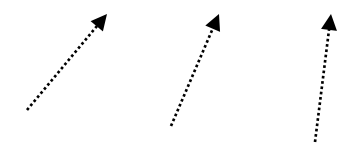
Salary

0-20K	1
20K-50K	2
50K- $\infty$	3

Grid

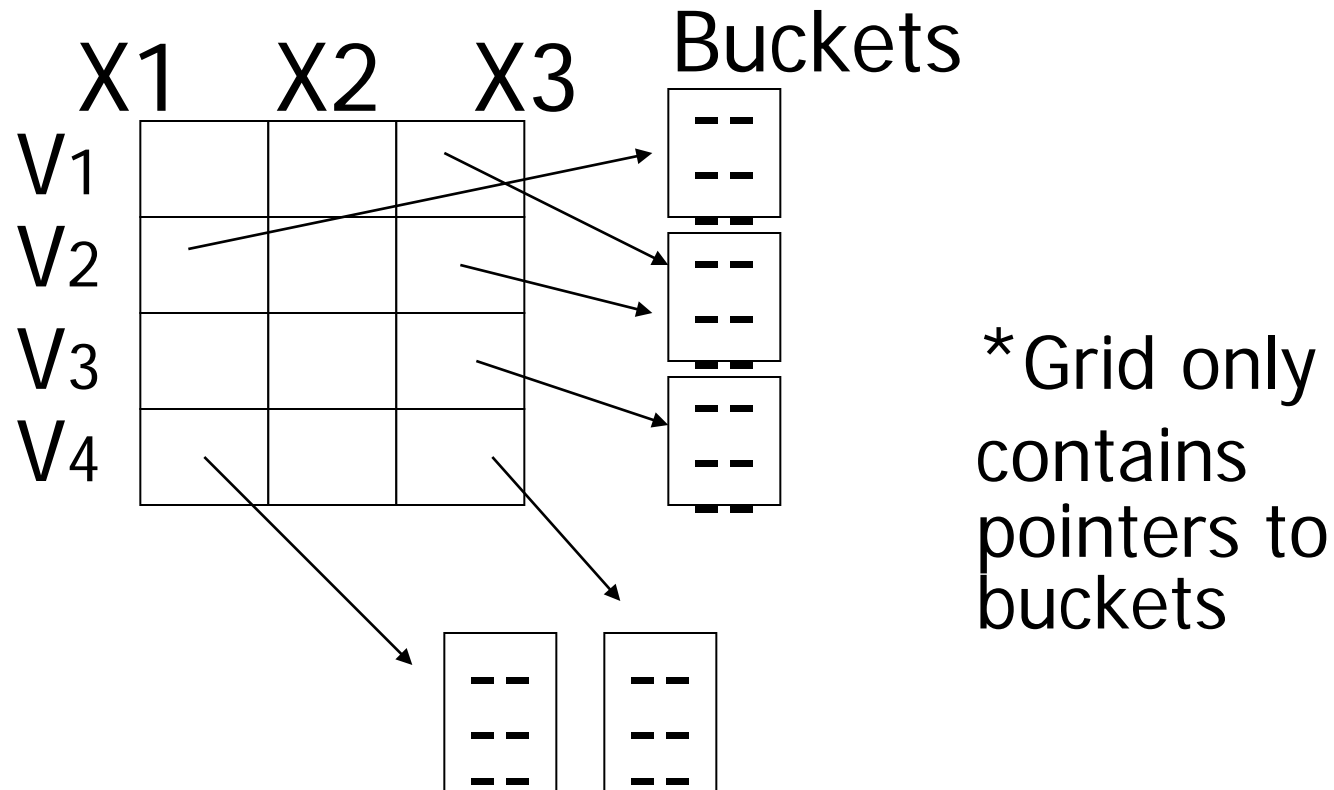
1	2	3
Toy	Sales	Personnel



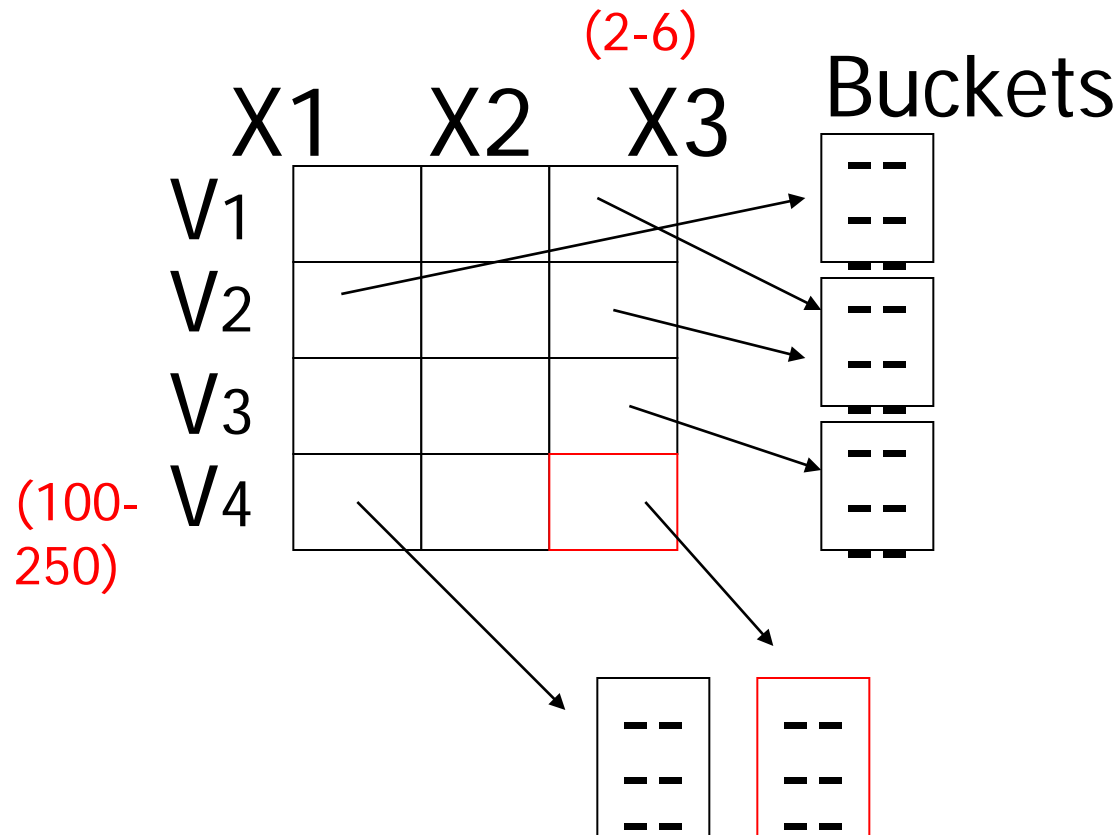
## CLAIM

- Can quickly find records with
  - key 1 =  $V_i$   $\wedge$  Key 2 =  $X_j$
  - key 1 =  $V_i$
  - key 2 =  $X_j$
- And also ranges....
  - E.g., key 1  $\geq V_i$   $\wedge$  key 2  $< X_j$

# Grid Index with Indirection



# Grid Index with Indirection - lookup

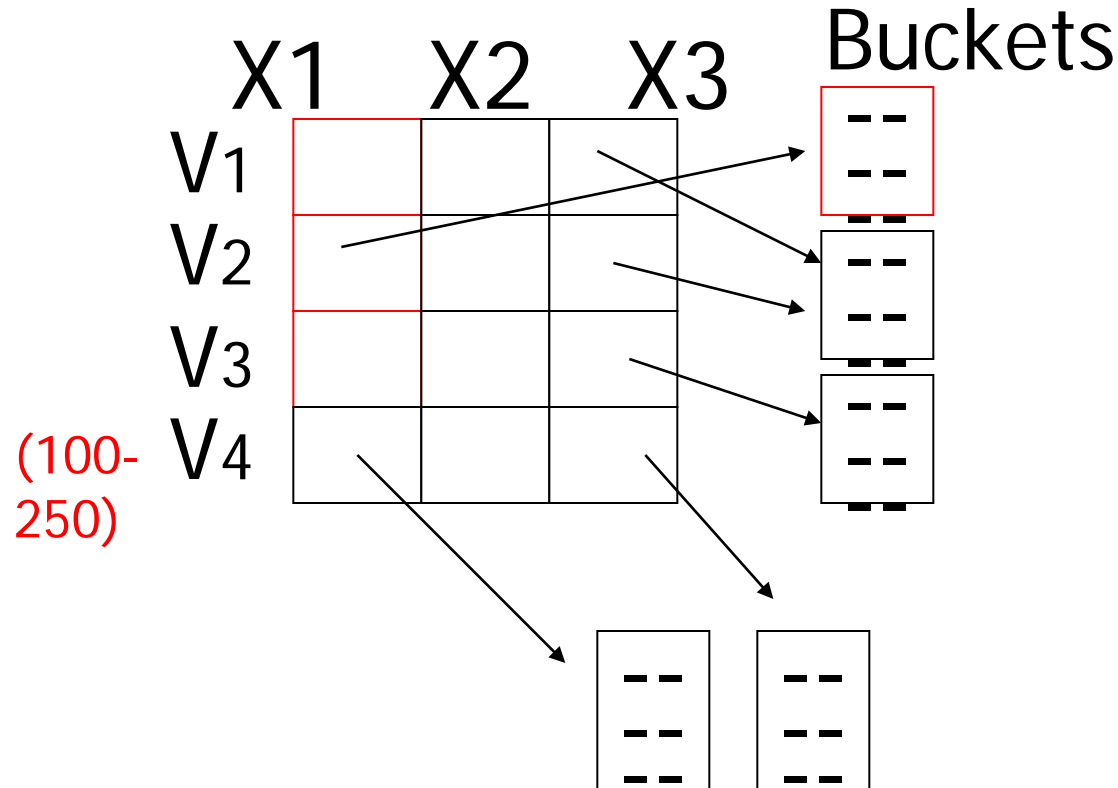


1) Lookup of Specific point-pair

e.g.  $X = 5, V = 200$

→ `Grid[V4][X3]`

# Grid Index with Indirection - lookup

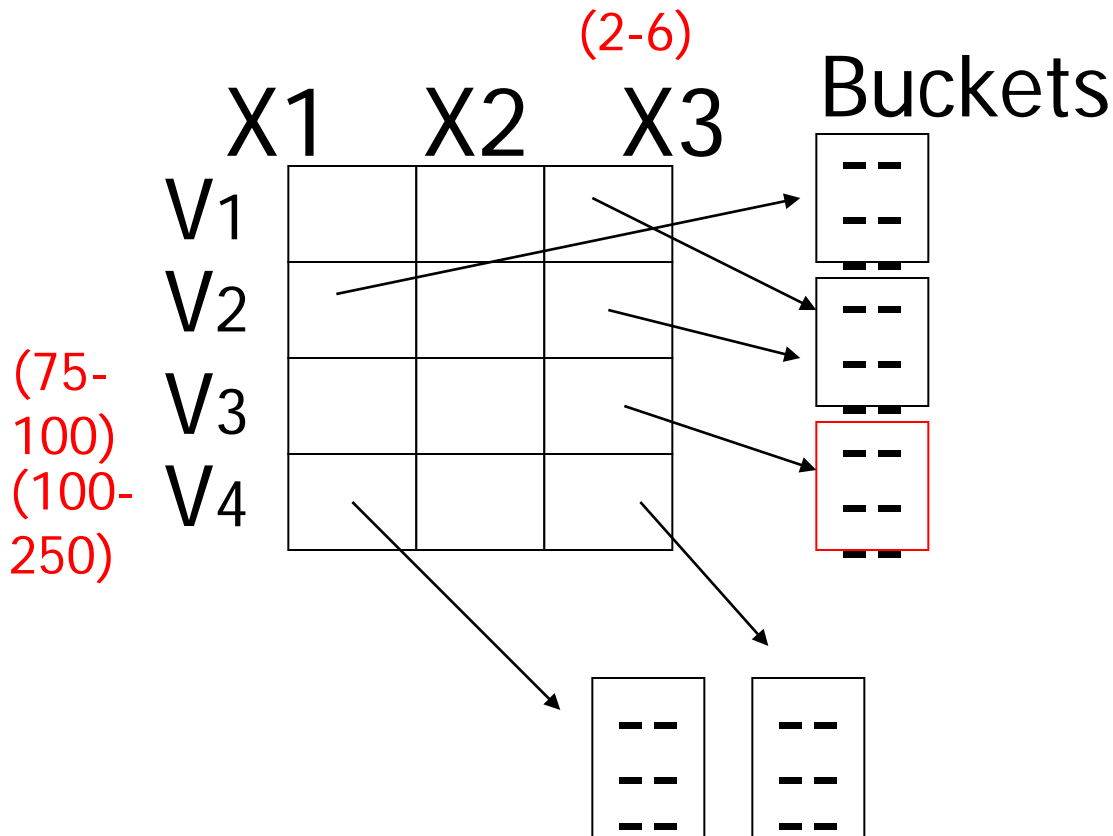


2) Partial Match Queries

e.g.  $V < 100$

→ Grid[V1][X1] ^  
 Grid[V2][X1] ^  
 Grid[V3][X1]

# Grid Index with Indirection - lookup

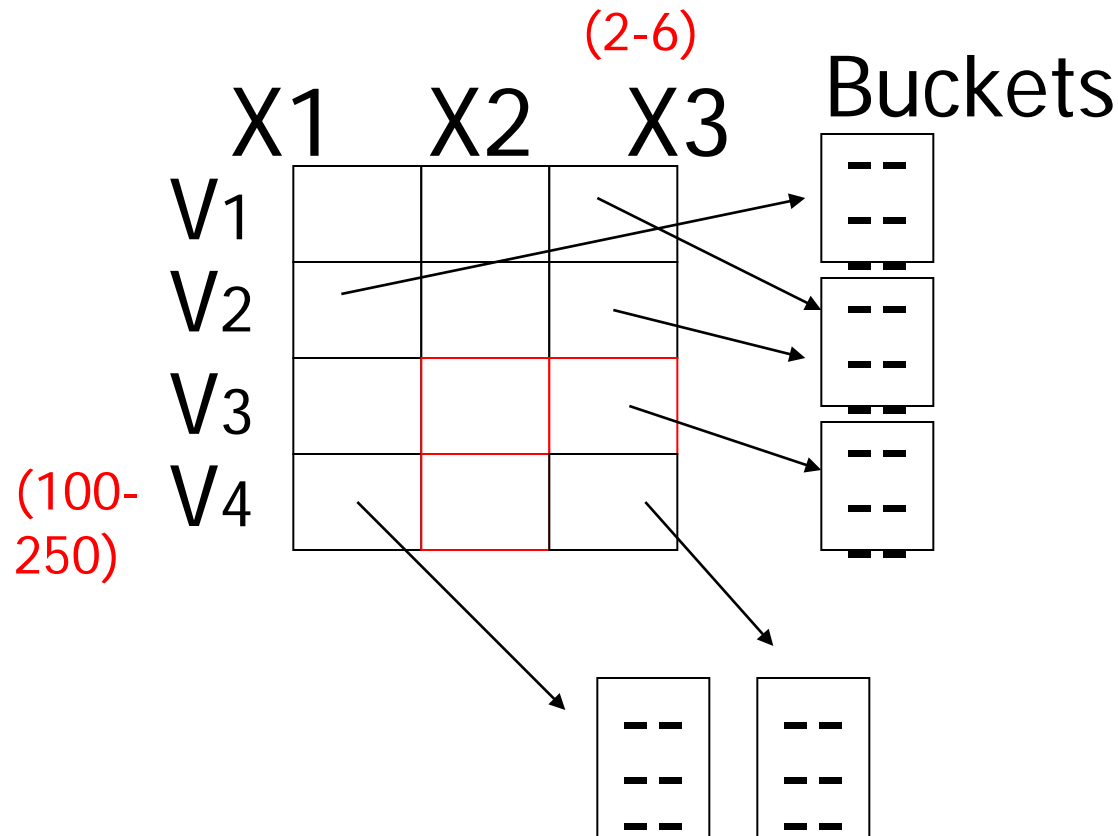


## 3) Range Queries

e.g.  $75 \leq V < 100$   
 $\&\& X \geq 4$

→ Grid[V3][X3]

# Grid Index with Indirection - lookup

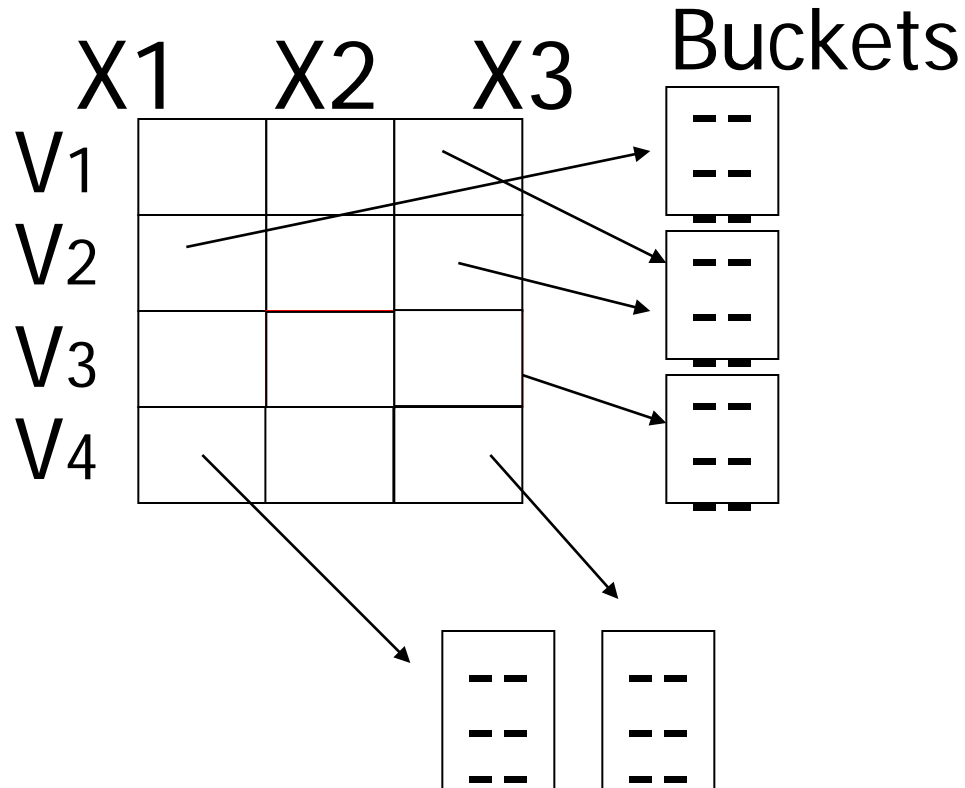


4) Nearest Neighbor Queries

e.g.  $V = 120$   
 $\&\& X \geq 3$

→ Grid[V3][X2] ^  
 Grid[V3][X3] ^  
 Grid[V4][X2]

# Grid Index with Indirection - insertion



- A) Do a lookup
- B) Add to bucket (overflow)





## Grid files

- ⊕ Good for multiple-key search
- ⊖ Space management overhead  
(nothing is free)
- ⊖ Need partitioning ranges that evenly split keys or evenly distributed data!