

# LAYOUT FLEXIBILITY

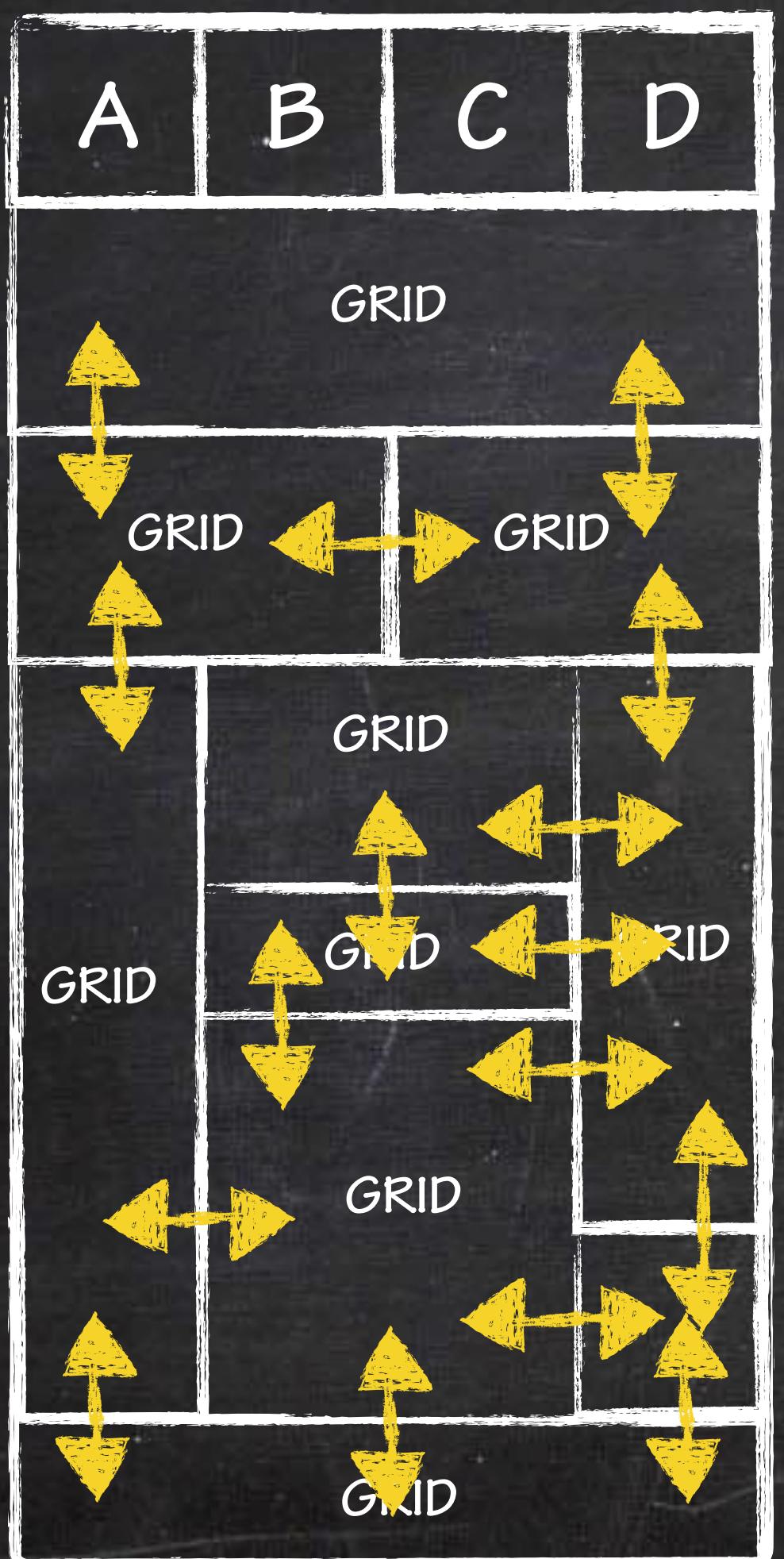
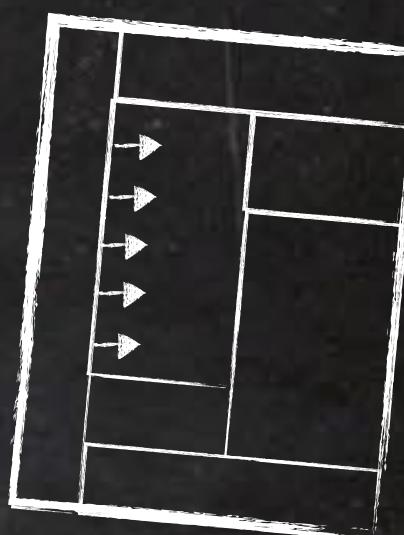
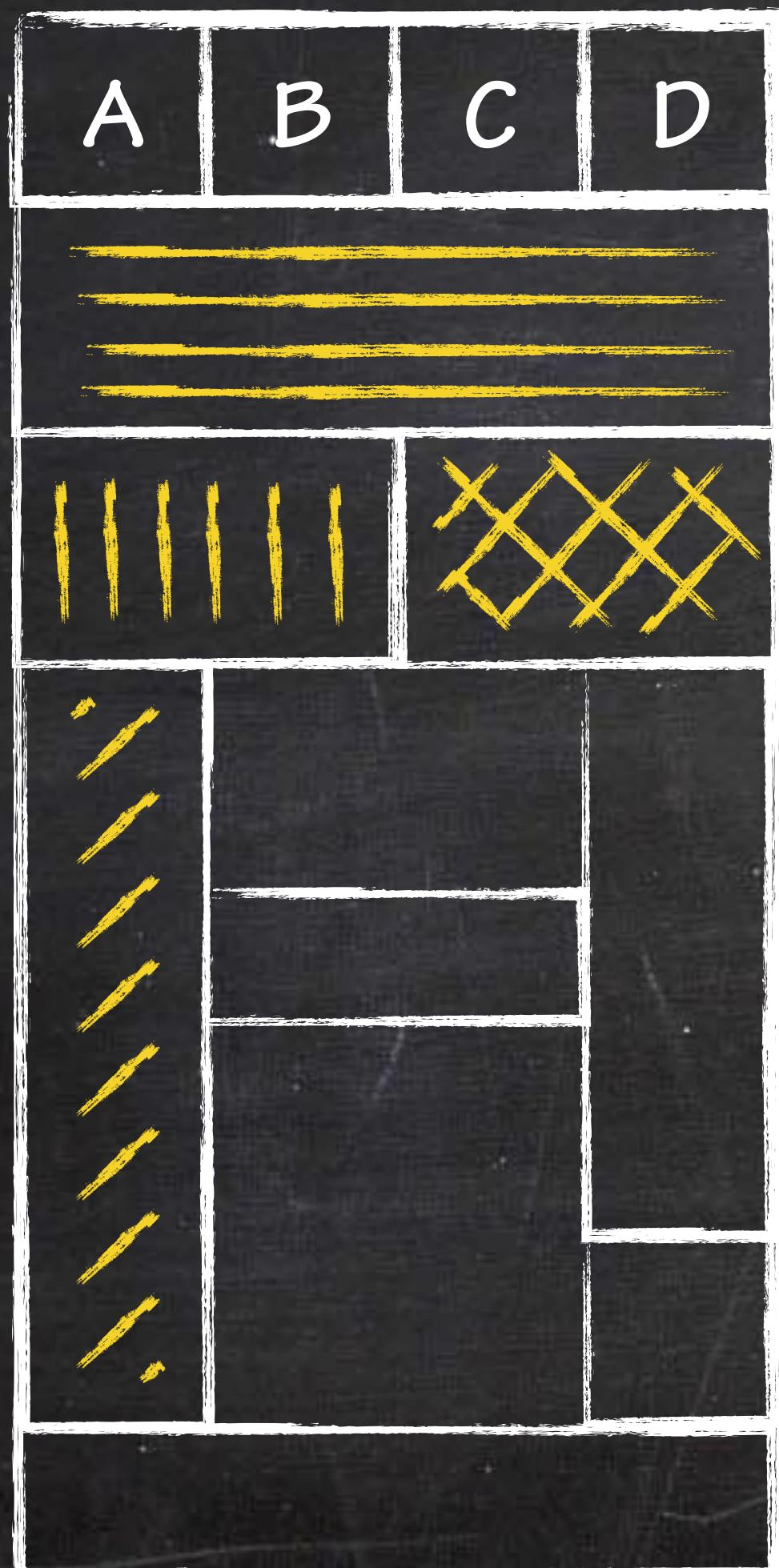


TABLE PARTITIONING  
**STRONG FLEXIBLE LAYOUT**



OUR SOLUTION  
**GRID TABLES**

# GRID-BASED PHYSICAL STORAGE

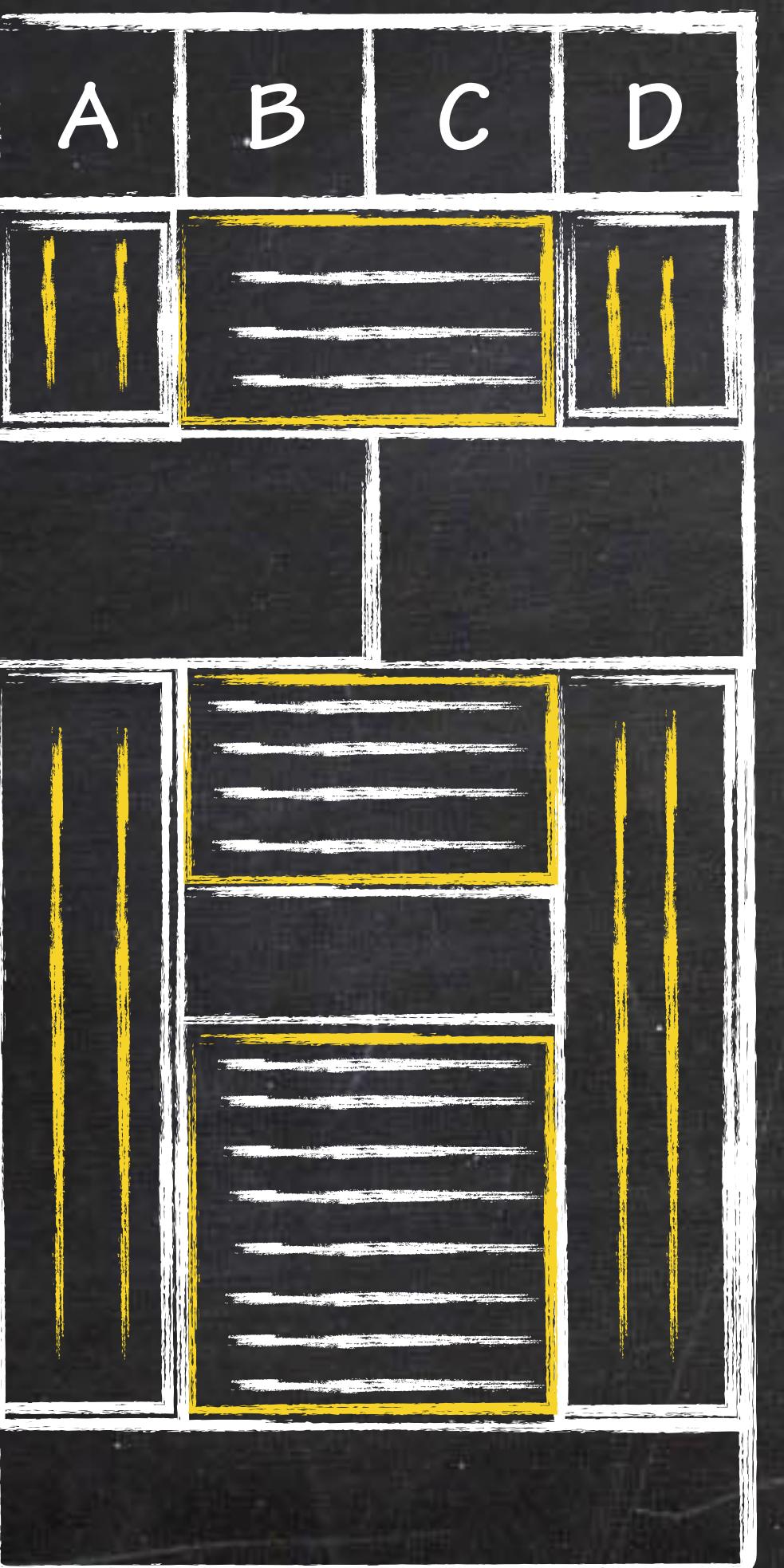


# OUR SOLUTION GRID TABLES

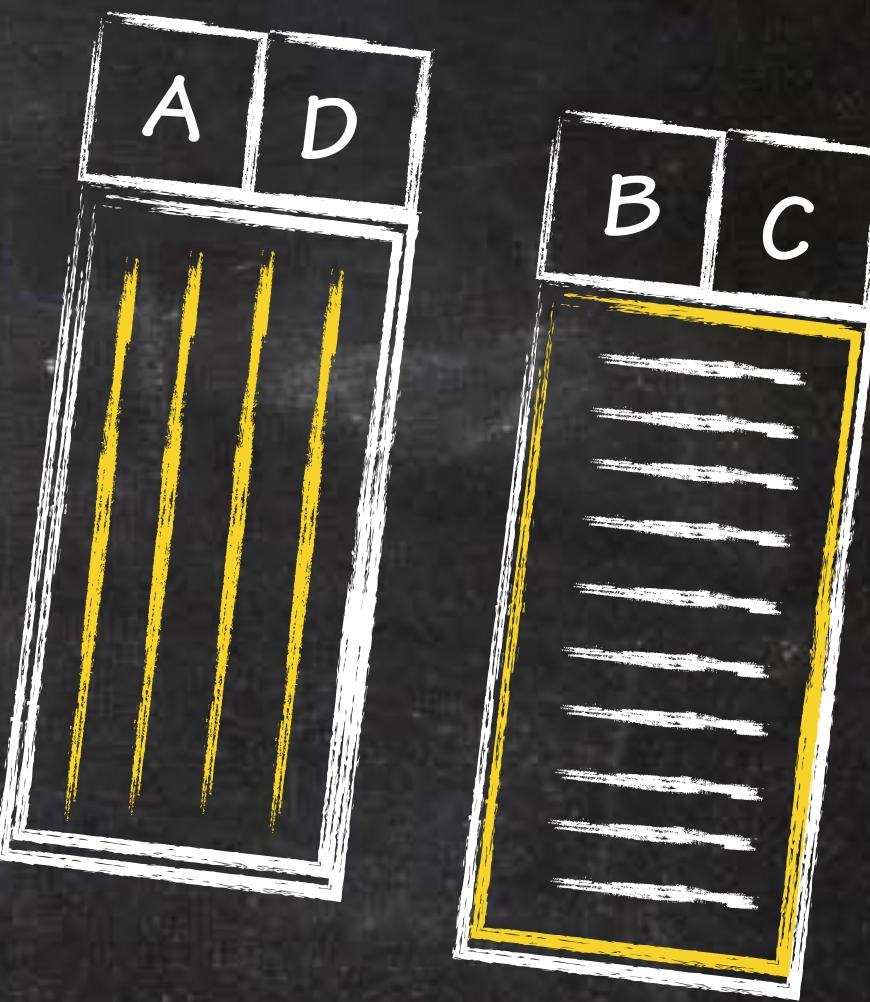
FINE-GRAIN PHYSICAL STORAGE  
PER-GRID RECORD ORGANIZATION



# RAW DATA PACKING

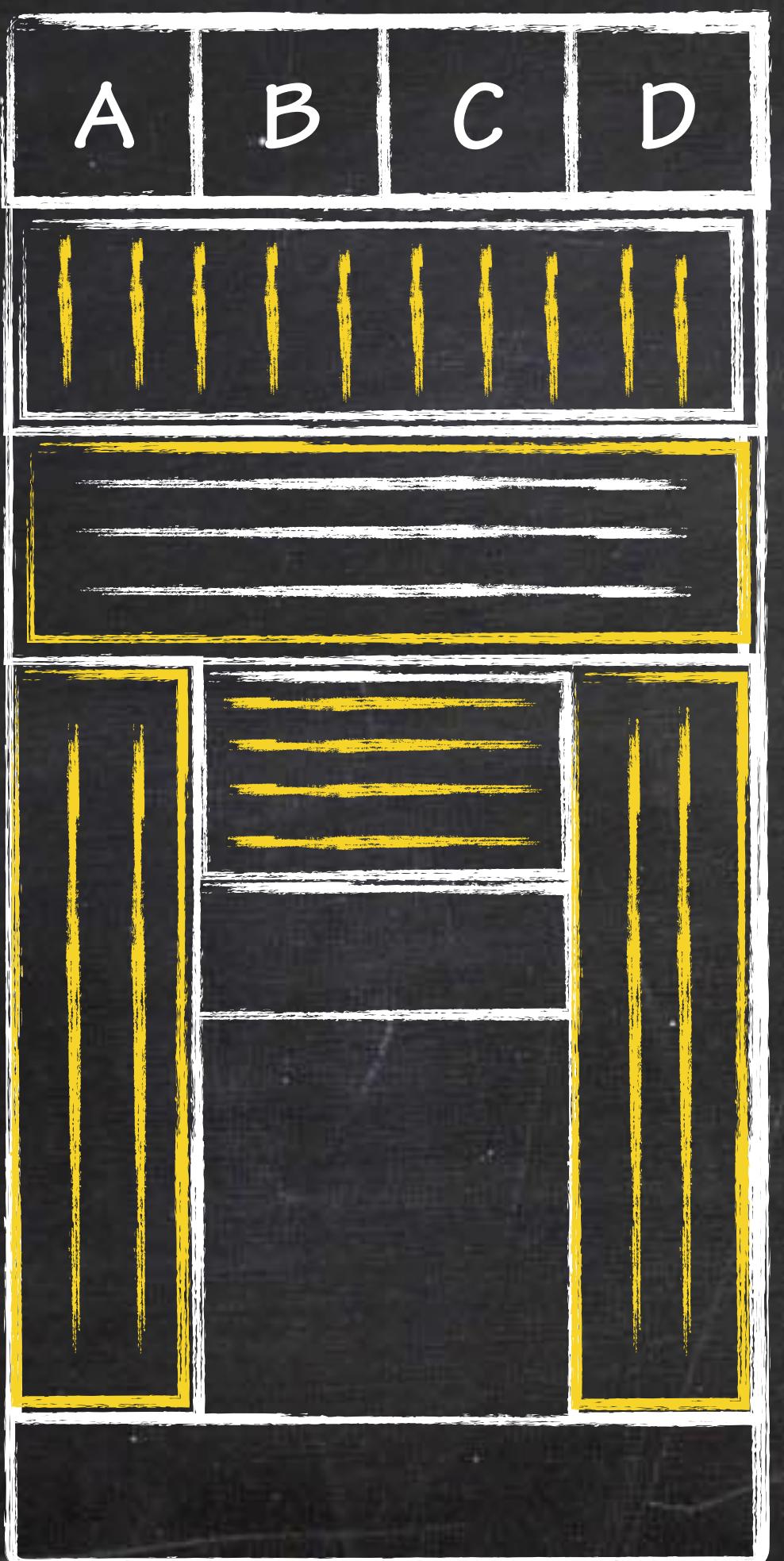


DATA CO-LOCATION  
PACKING OF GRID DATA FRAGMENTS

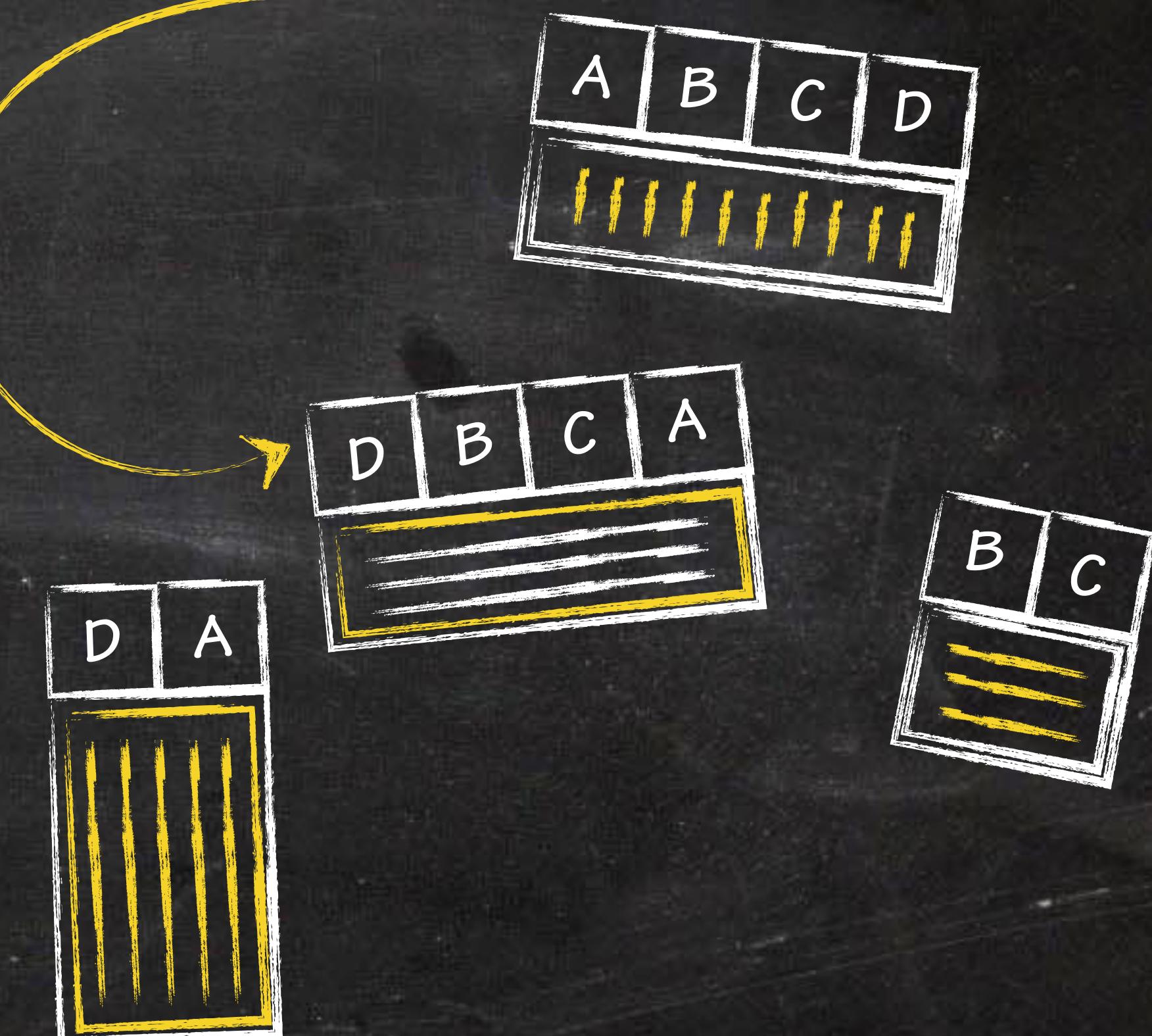


# OUR SOLUTION **GRID TABLES**

# SCHEMA RE-ORDERING



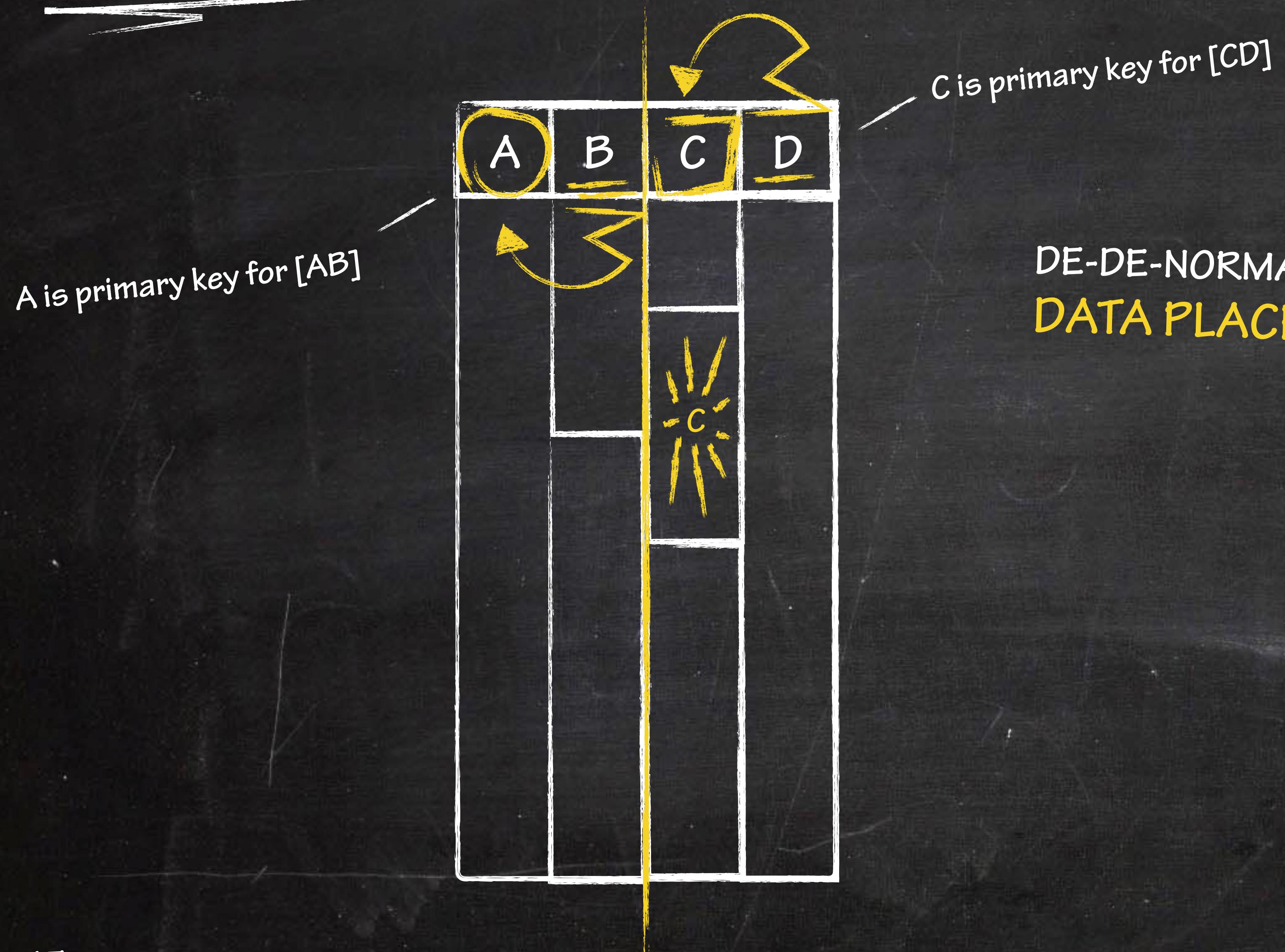
ENHANCED CACHE EFFICIENCY  
PER-GRID SCHEMA RE-ORDERING



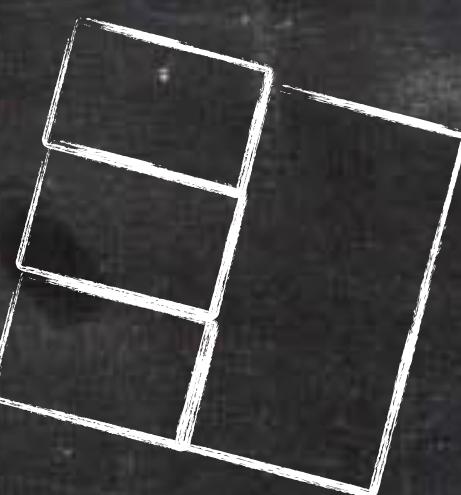
# OUR SOLUTION GRID TABLES

# DATA INTEGRITY RULES

# OUR SOLUTION GRID TABLES



DE-DE-NORMALIZE  
DATA PLACEHOLDERS & INTEGRITY



# SPARSITY AS FIRST CLASS CITIZEN



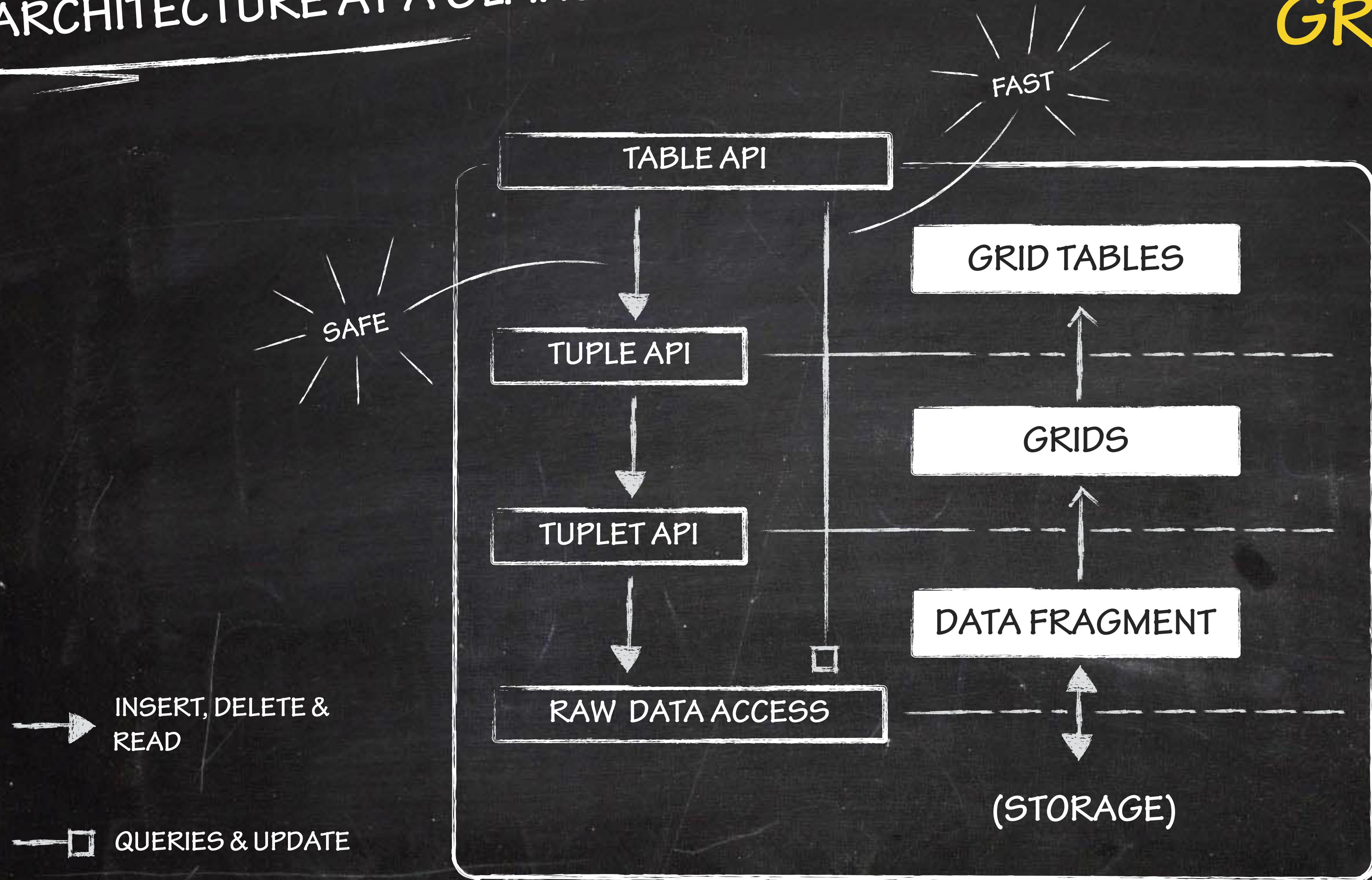
# OUR SOLUTION GRID TABLES

NULL VALUE SUPPRESSION  
**NO MEMORY TO ENCODE NULLS**

**NO GRID, NO VALUE!**

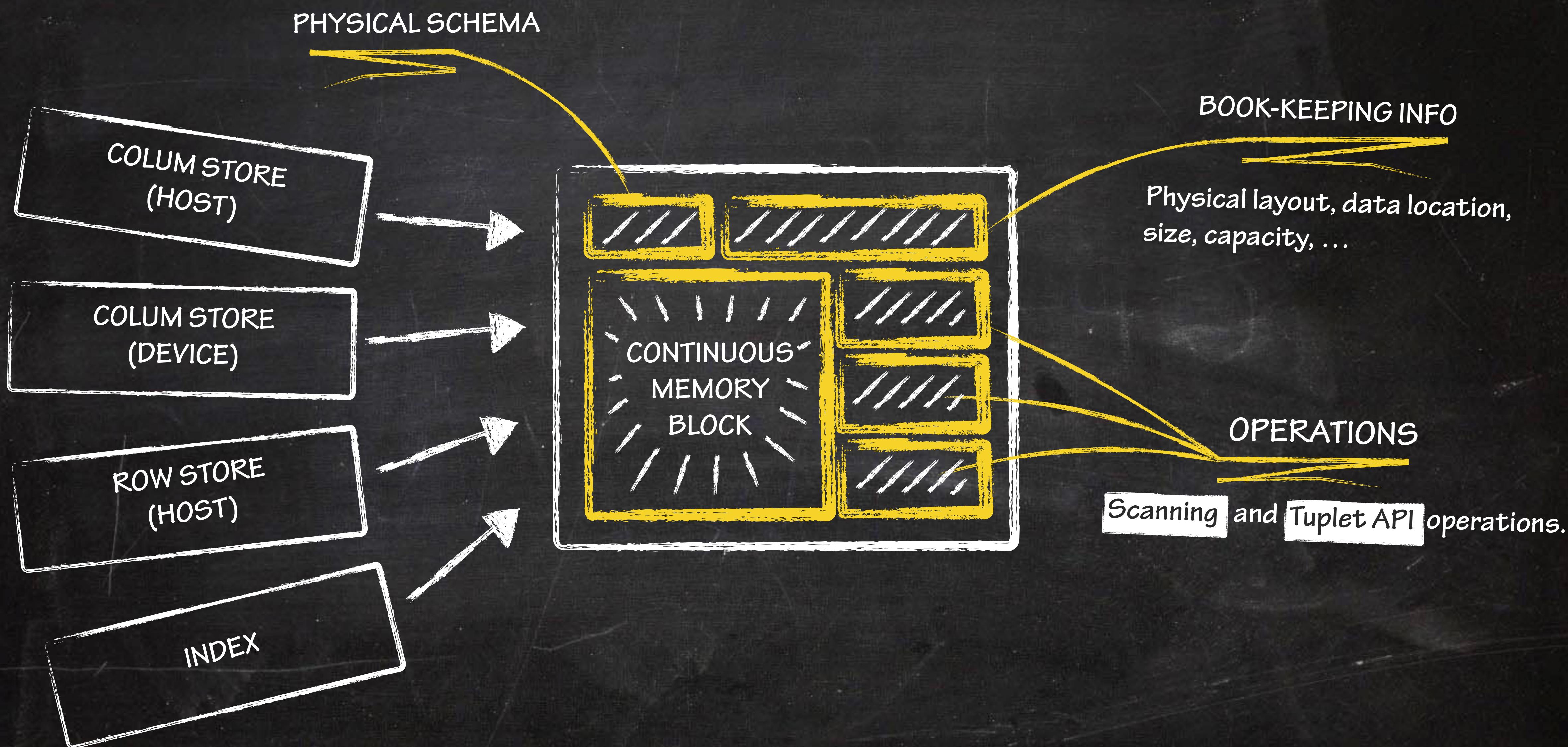
# ARCHITECTURE AT A GLANCE

# OUR SOLUTION GRID TABLES



# DATA FRAGMENTS

# OUR SOLUTION GRID TABLES



# OUR SOLUTION GRID TABLES

GRIDS

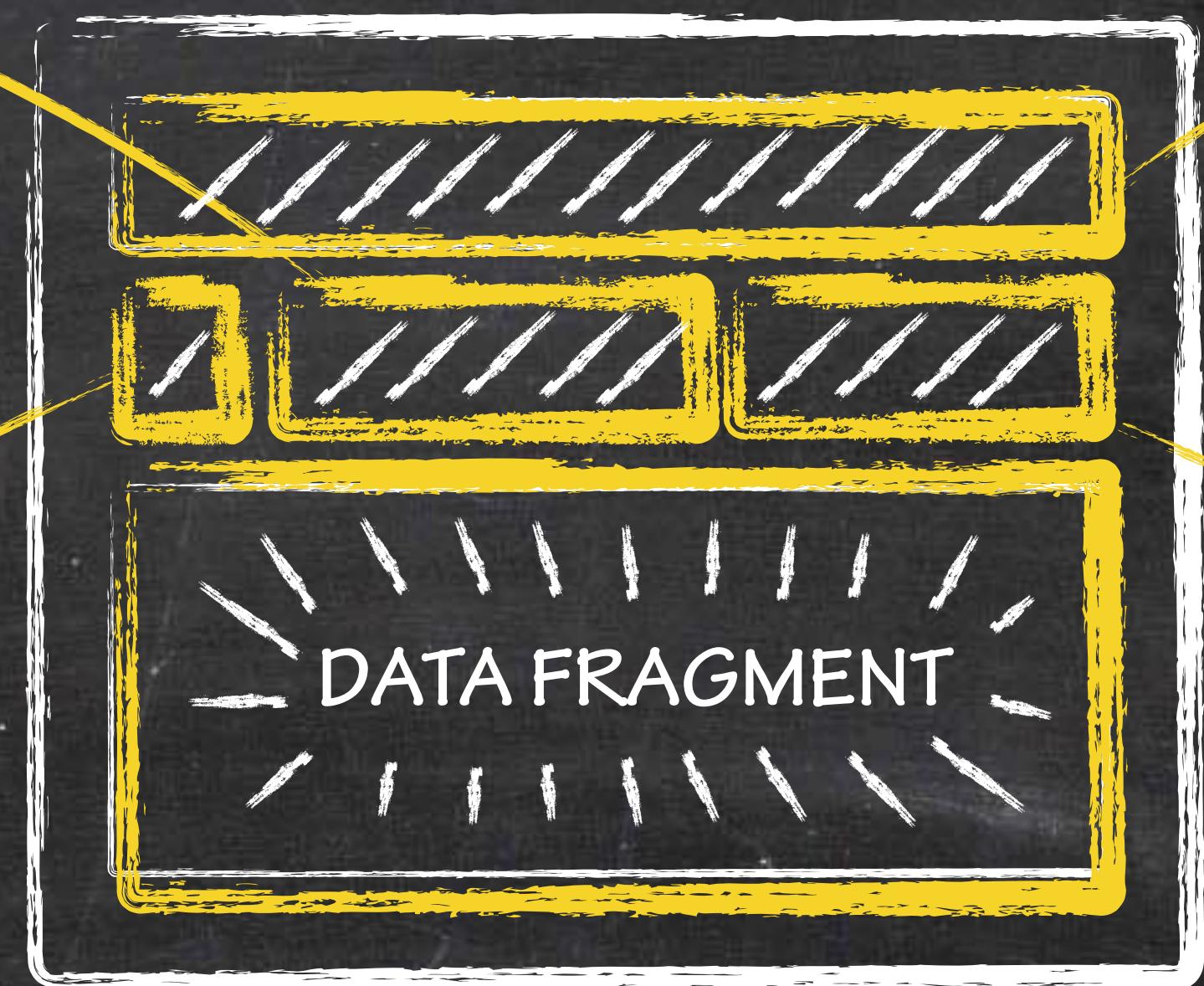
TUPLE ID COVER

List of tuple id  
contributes data

this grid

MUTEX LOCK

Prevention of  
concurrent writes.



Direct access from table  
to data fragment  
index mapping.  
via

via

CACHE

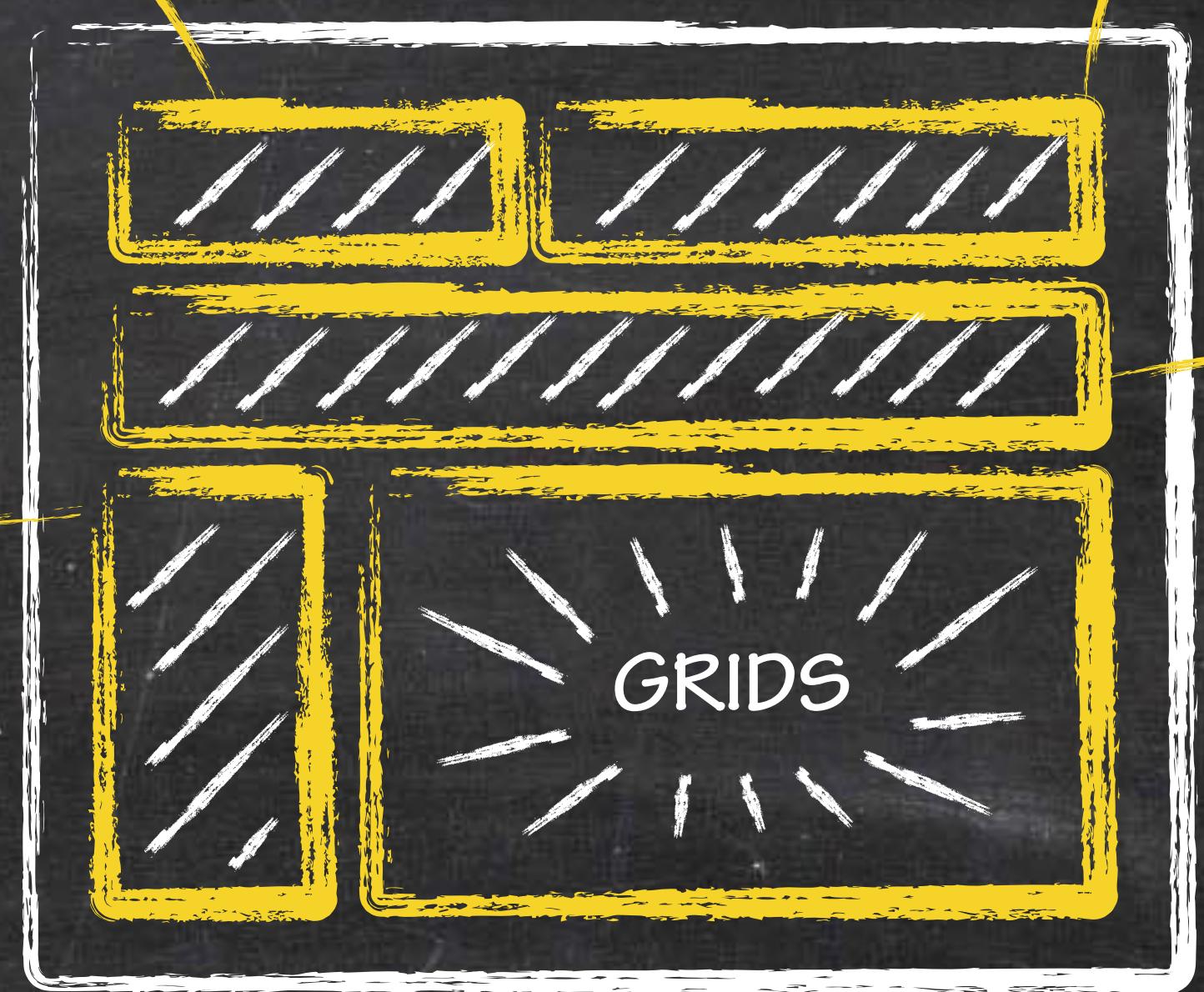
Caches last accesses to  
tuple id cover for translation  
to .

# GRID TABLES

## TABLE (LOGICAL) SCHEMA



Maps intervals of grids that contribute to the tuple to list of



RANGE QUERIES  
ON TUPLE IDS

# OUR SOLUTION GRID TABLES

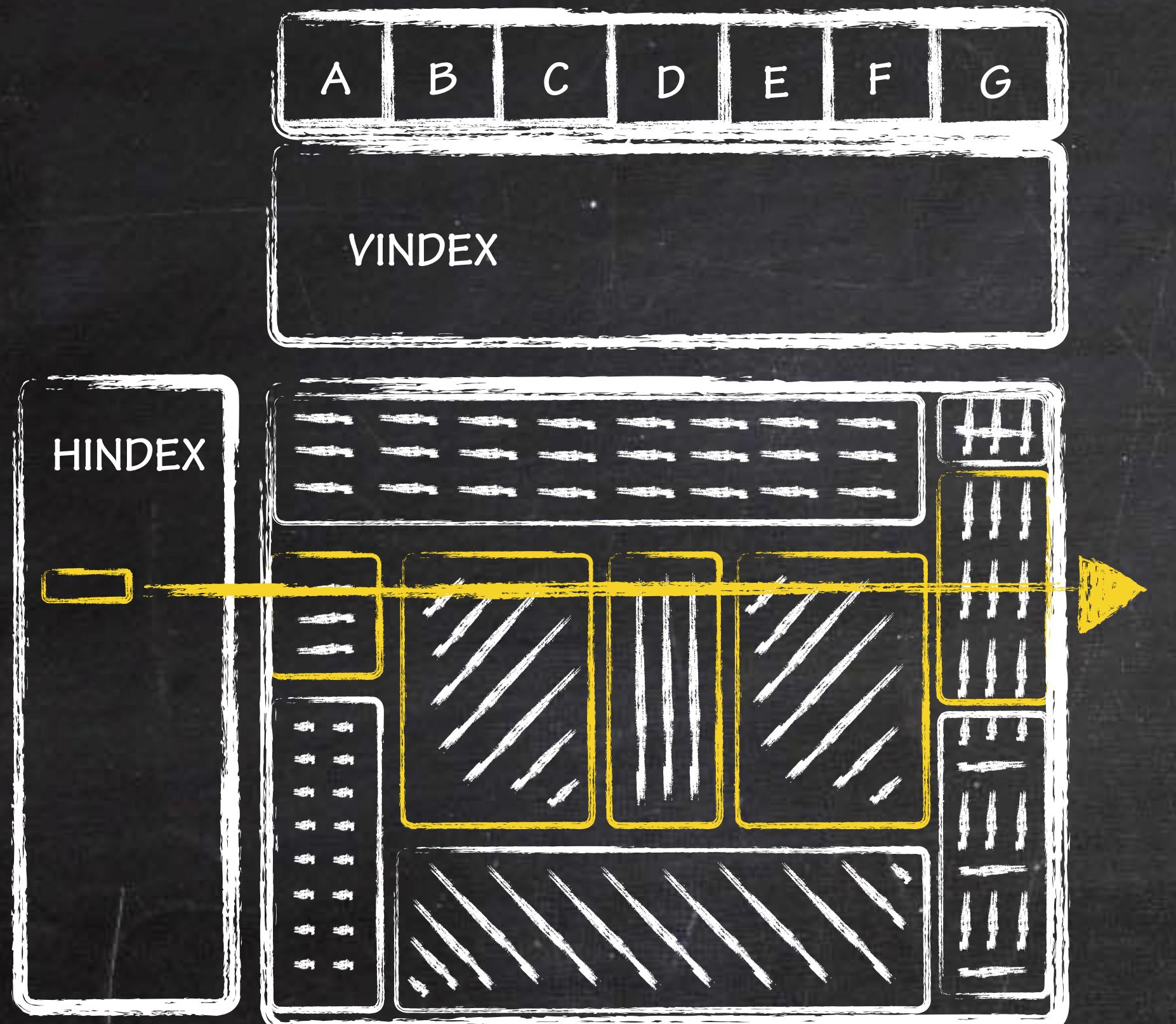
## BOOK-KEEPING INFO



Maps each table schema to list of grids that cover that attribute.

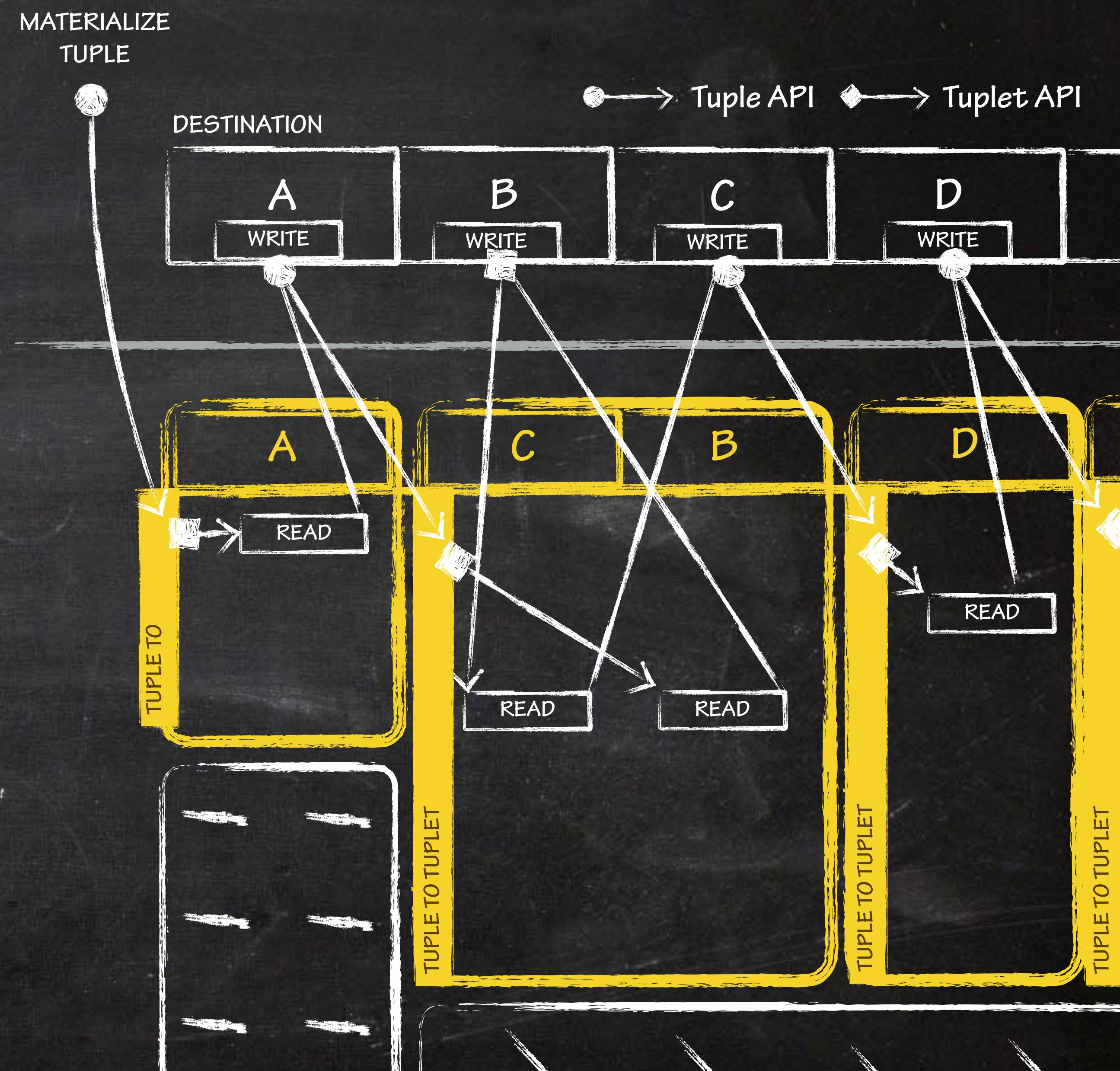
POINT QUERIES ON ATTRIBUTES

# GRID TABLE OPERATIONS

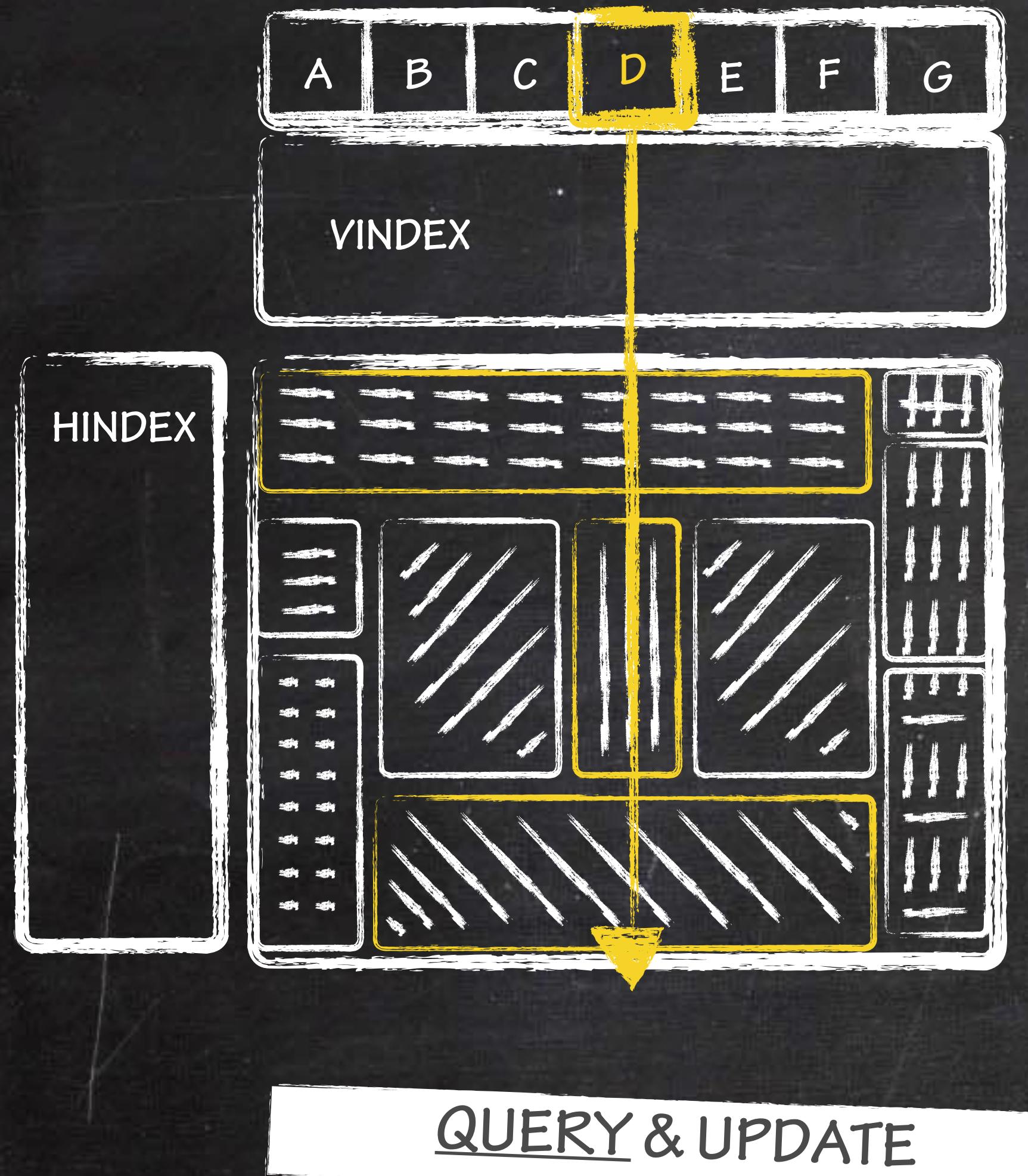


Tuple API abstracts from low-level seeking (e.g., per-grid schema re-ordering), and tuple id to per-grid tuplet id conversion.

# OUR SOLUTION GRID TABLES



# GRID TABLE OPERATIONS



Per-grid raw data access for maximal performance. After per-grid query, tuplet id to tuple id conversion is executed.

# OUR SOLUTION GRID TABLES

