

# QUERY/STORAGE OPTIMIZER OVER-THE-EDGE OPTIMIZATION

QUERY ENGINE

RESOURCE UTILIZATION & TXN SEMANTICS

STORAGE ENGINE OPTIMIZER

ACCESS-PATTERN-SPECIFIC  
SELF-MODIFICATIONS AND SELF-OPTIMIZATIONS

STORAGE ENGINE

RECORD STORAGE FOR OLTP & OLAP

## RUNNING EXAMPLE

CUSTOMER

<u>CID</u>	CNAME	ADDRESS
1	N <sub>1</sub>	A <sub>1</sub>
2	N <sub>1</sub>	A <sub>2</sub>
3	N <sub>2</sub>	A <sub>3</sub>
4	N <sub>3</sub>	A <sub>3</sub>

BUY

<u>CID</u>	<u>PID</u>	AMOUNT
1	2	N <sub>1</sub>
1	4	N <sub>2</sub>
2	2	N <sub>3</sub>
3	1	N <sub>4</sub>

PRODUCT

<u>PID</u>	PNAME	QUANTITY
1	P <sub>1</sub>	Q <sub>1</sub>
2	P <sub>2</sub>	Q <sub>2</sub>
3	P <sub>3</sub>	Q <sub>3</sub>
4	P <sub>4</sub>	Q <sub>4</sub>

## ... PHYSICAL LAYOUT

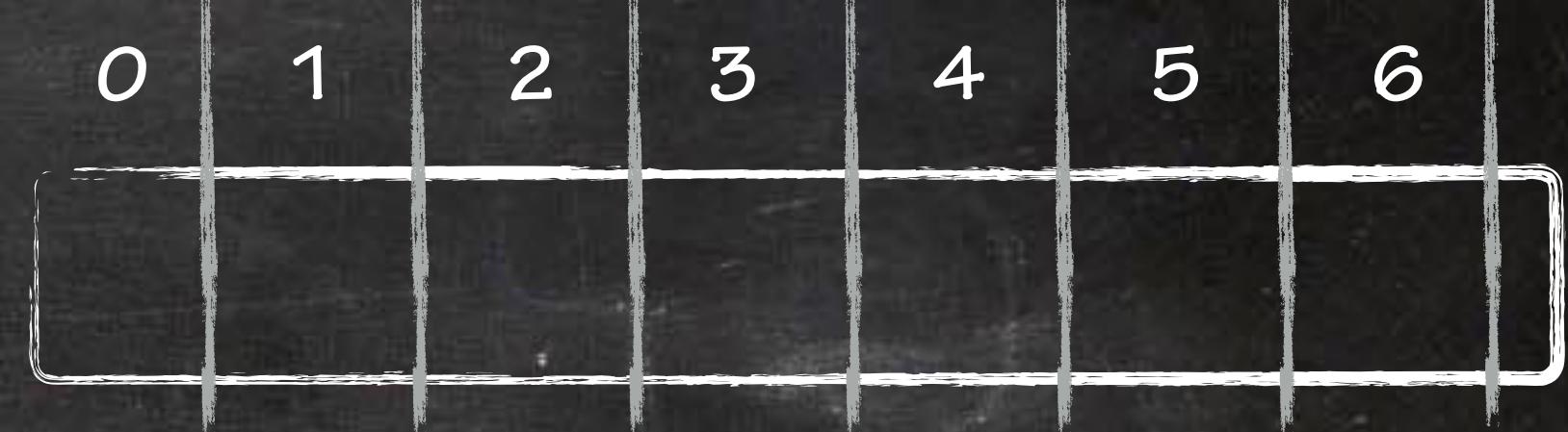
A	B	C	D
$a_1$	$b_1$	$c_1$	$d_1$
$a_2$	$b_2$	$c_2$	$d_2$
$a_3$	$b_3$	$c_3$	$d_3$
$a_4$	$b_4$	$c_4$	$d_4$

LOGICAL  
RELATION (2D)

LINEARIZATION

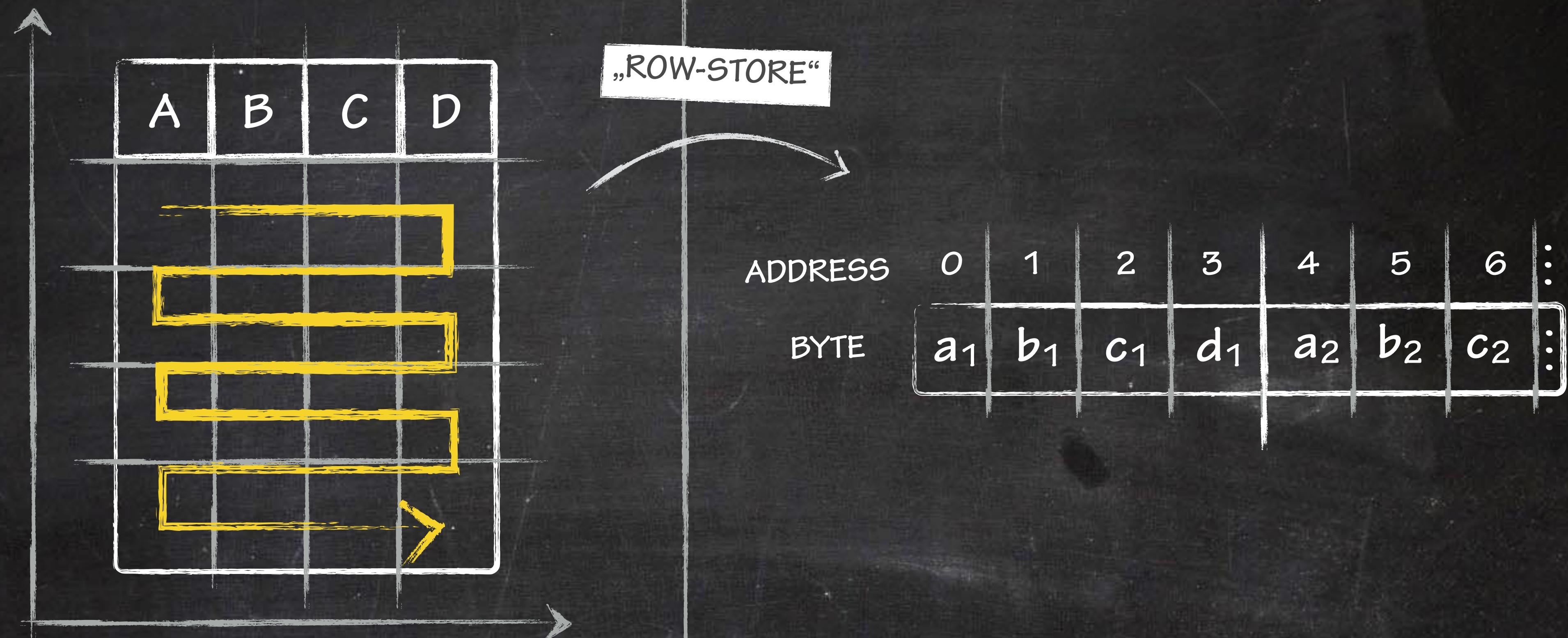
ADDRESS

BYTE

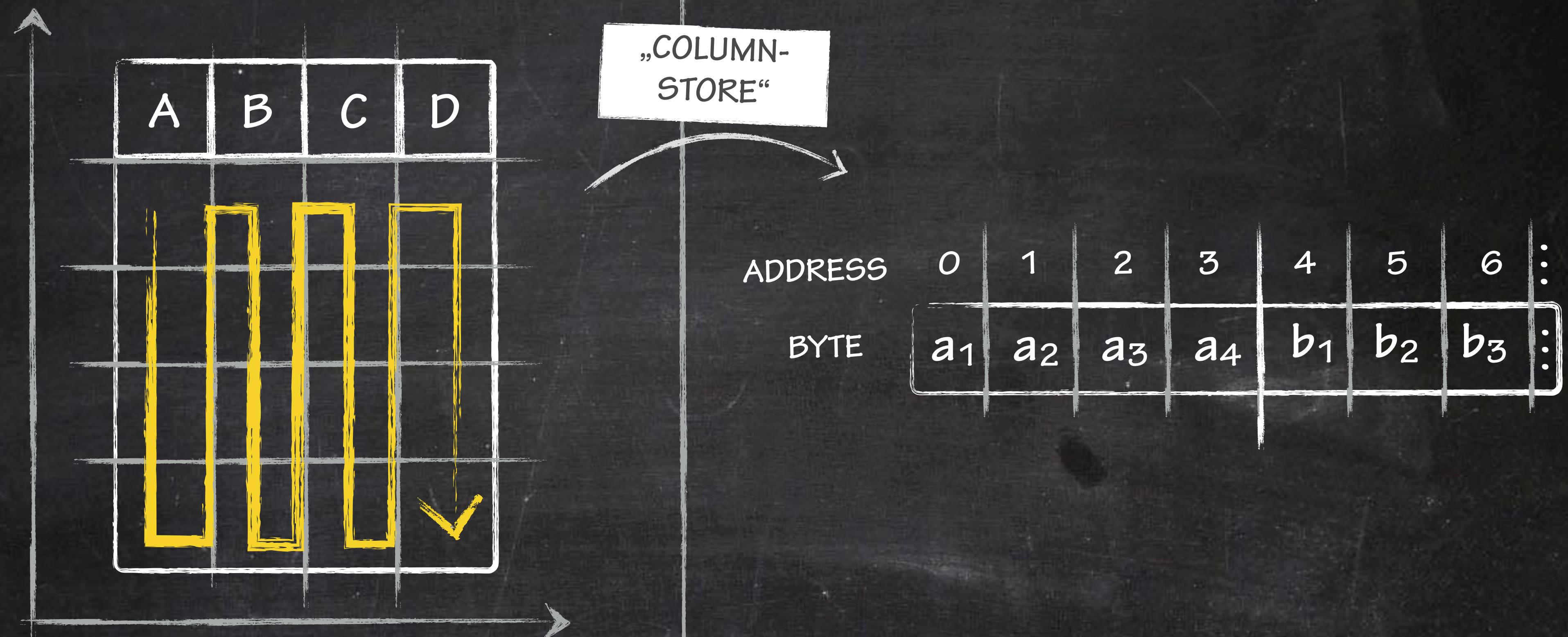


PRIMARY MEMORY IS LOCATION-  
ADDRESSED (1D)

## ... PHYSICAL LAYOUT

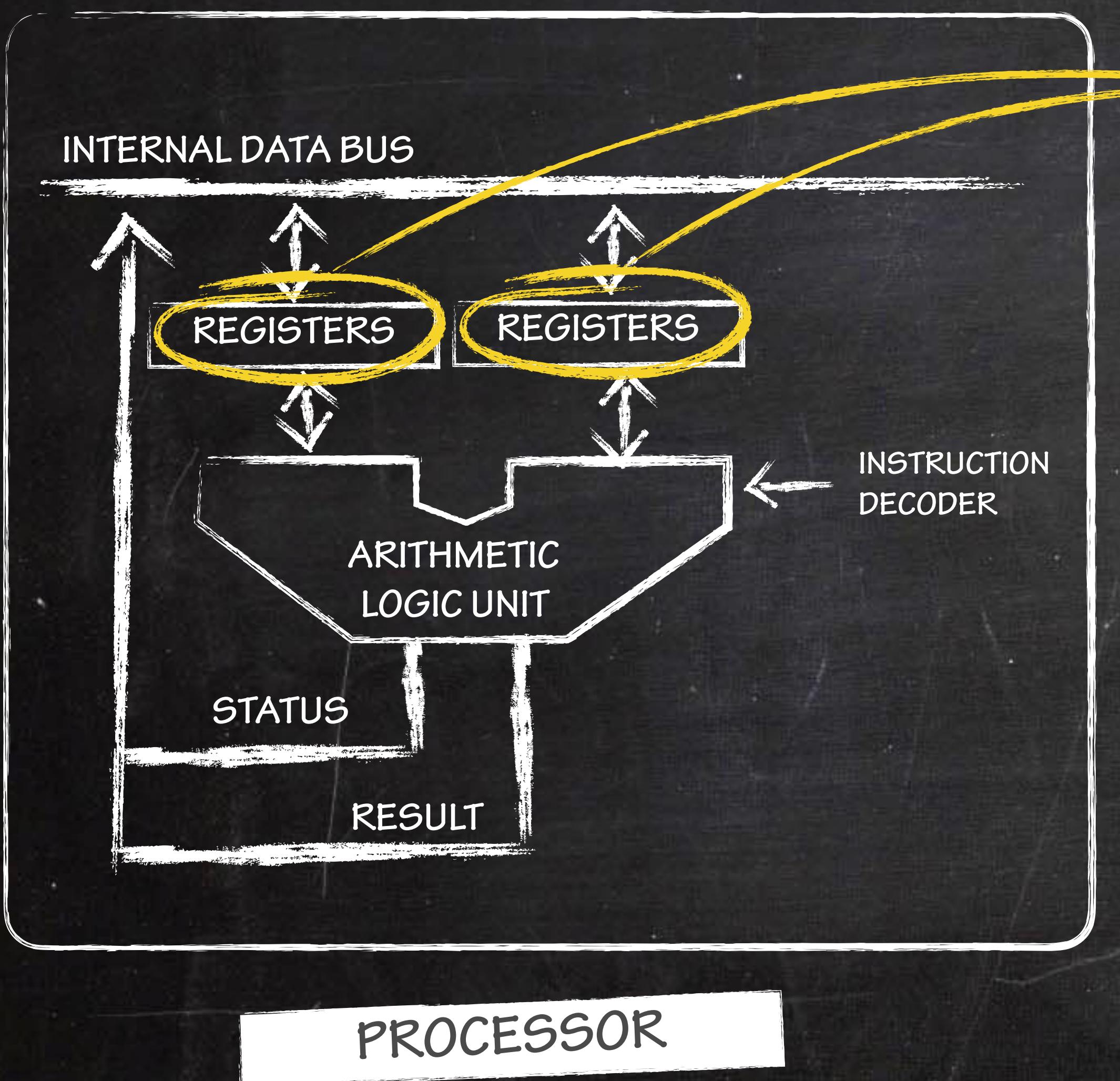


## ... PHYSICAL LAYOUT

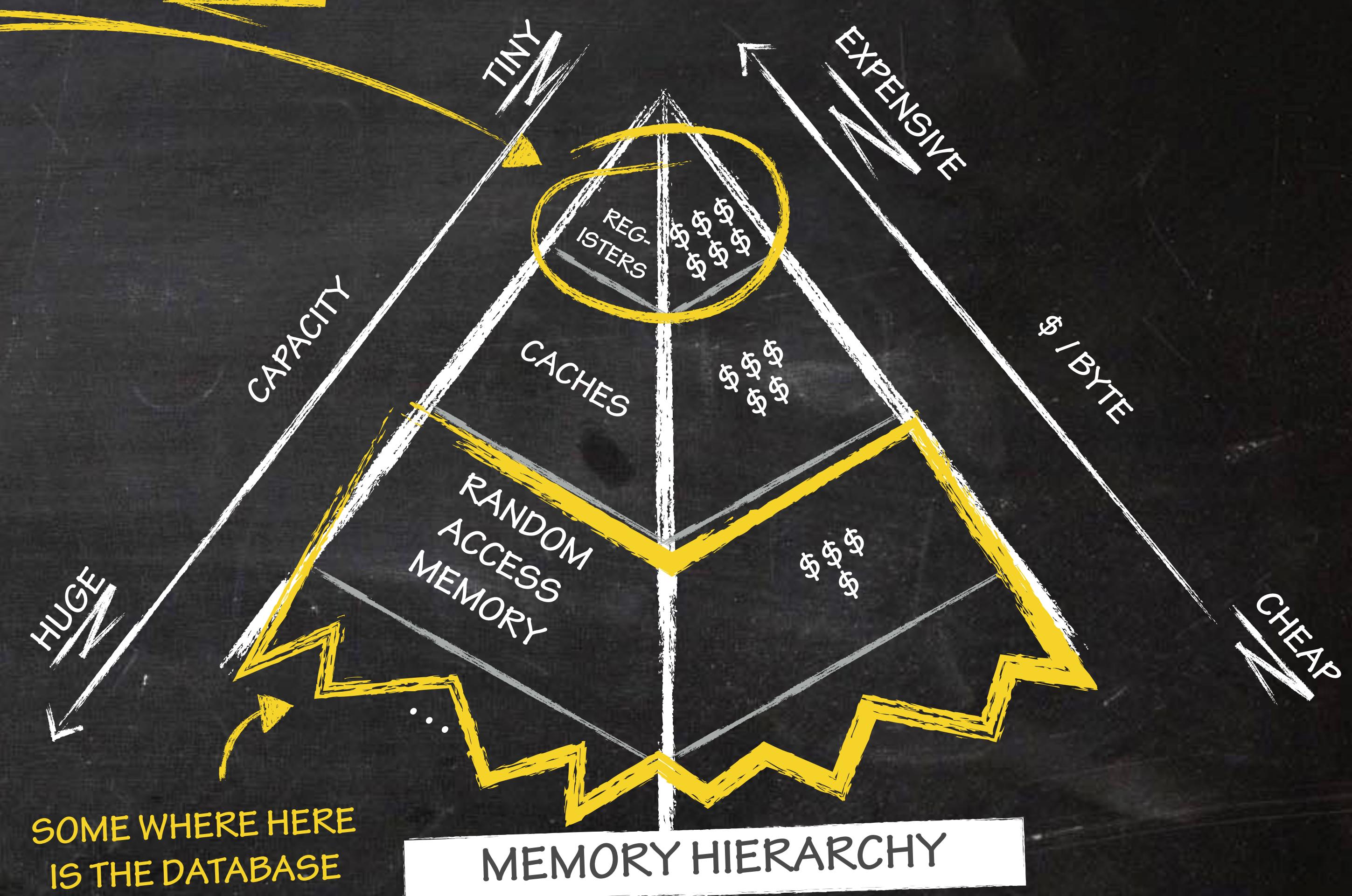


# WHY DO WE CARE?

... PHYSICAL LAYOUT

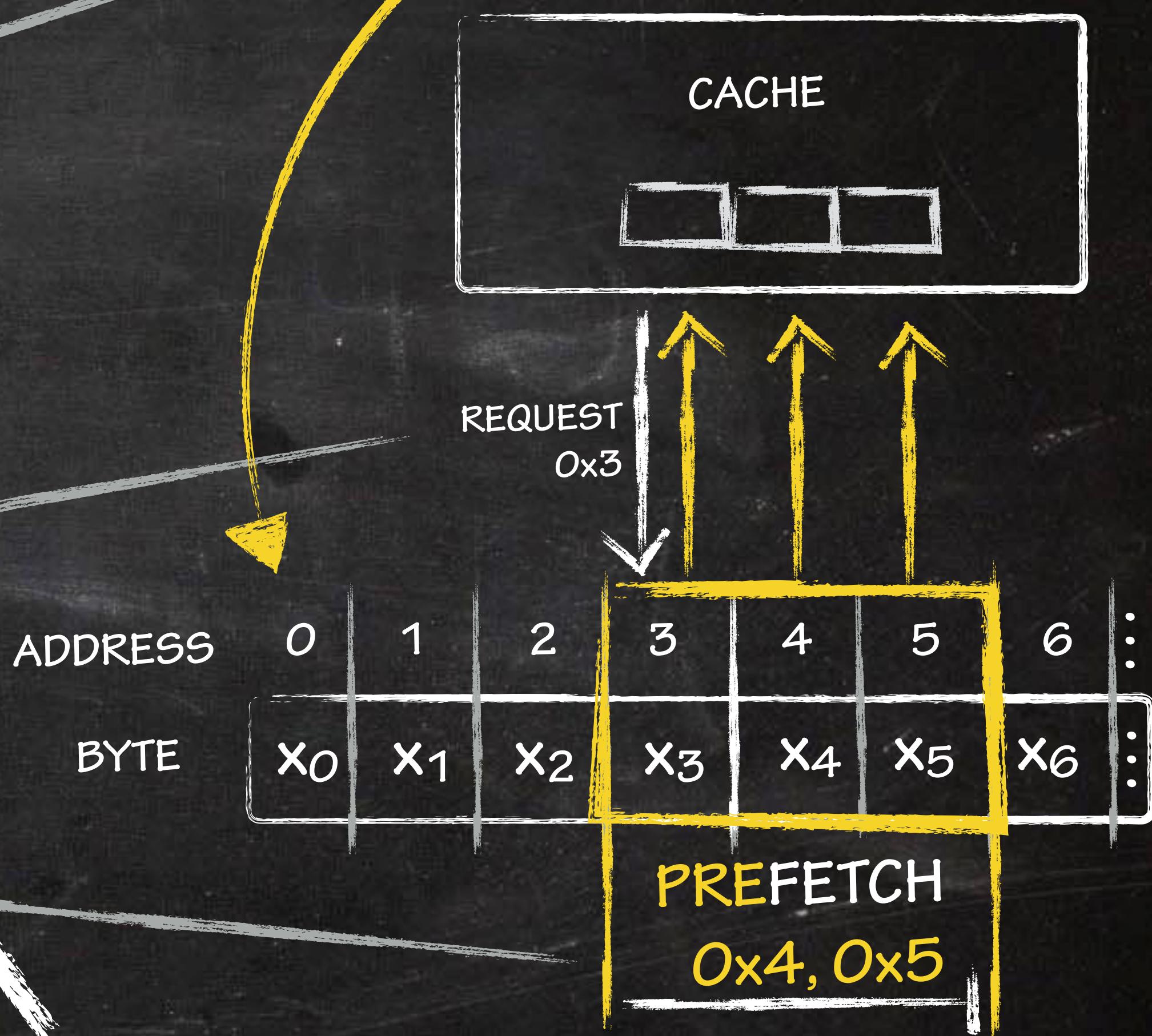
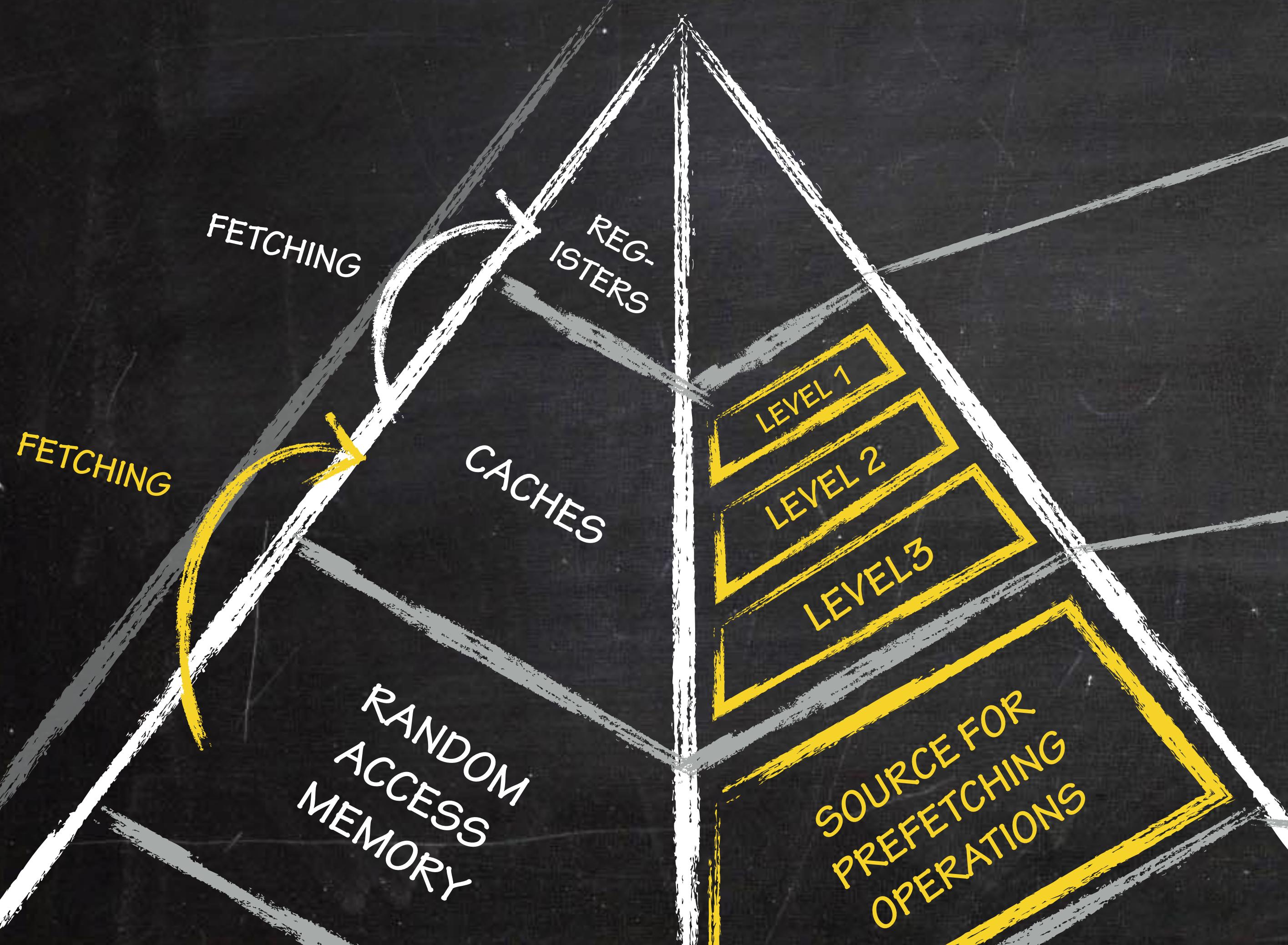


THE ONLY PLACE THE  
PROCESSOR CAN ACCESS DATA



# WHY DO WE CARE? PREFETCHING!

... PHYSICAL LAYOUT



# OLTP access pattern

CUSTOMER

<u>CID</u>	CNAME	ADDRESS
1	N <sub>1</sub>	A <sub>1</sub>
2	N <sub>1</sub>	A <sub>2</sub>
4	N <sub>3</sub>	A <sub>3</sub>

BUY

<u>CID</u>	<u>PID</u>	AMOUNT
1	2	N <sub>1</sub>
2	4	N <sub>2</sub>
	2	N <sub>3</sub>
	1	N <sub>4</sub>

PRODUCT

<u>PID</u>	PNAME	QUANTITY
1	P <sub>1</sub>	Q <sub>1</sub>
2	P <sub>2</sub>	Q <sub>2</sub>
3	P <sub>3</sub>	Q <sub>3</sub>
4	P <sub>4</sub>	Q <sub>4</sub>

QOLTP

$$\pi_{\text{PNAME}} \left( \pi_{\text{PID}} \left( \sigma^{\text{INDEX}}_{\text{CID} = 3} (\text{CUSTOMER}) \right) \bowtie^{\text{HASH}_{\text{EQ}(\text{CID})}} (\text{BUY}) \right) \bowtie^{\text{HASH}_{\text{EQ}(\text{PID})}} (\text{PRODUCT})$$

## NORMALIZED TABLES

## OLTP Optimized



# OLAP access pattern

CUSTOMER			BUY			PRODUCT		
<u>CID</u>	CNAME	ADDRESS	<u>CID</u>	<u>PID</u>	AMOUNT	<u>PID</u>	PNAME	QUANTITY
1	N <sub>1</sub>	A <sub>1</sub>		2	N <sub>1</sub>	1	P <sub>1</sub>	Q <sub>1</sub>
2	N <sub>1</sub>	A <sub>2</sub>		4	N <sub>2</sub>	2	P <sub>2</sub>	Q <sub>2</sub>
3	N <sub>2</sub>	A <sub>3</sub>	2	2	N <sub>3</sub>	3	P <sub>3</sub>	Q <sub>3</sub>
4	N <sub>3</sub>	A <sub>3</sub>	3	1	N <sub>4</sub>	4	P <sub>4</sub>	Q <sub>4</sub>

QOLAP

$\gamma_{\text{SUM, QUANTITY}}^{\text{HASH}} \left( \left( \sigma_{\text{ADDRESS} = A_1}^{\text{SCAN}} (\text{CUSTOMER}) \right) \bowtie_{\text{EQ(CID)}}^{\text{HASH}} (\text{BUY}) \right) \bowtie_{\text{EQ(PID)}}^{\text{HASH}} (\text{PRODUCT})$

## DE-NORMALIZED TABLES

## OLAP Optimized

**WIDETABLE (CUSTOEMR, BUY, PRODUCT)**

CID	CNAME	ADDRESS	AMOUNT	PID	PNAME	QUANTITY
1	N <sub>1</sub>	A <sub>1</sub>	N <sub>1</sub>	2	P <sub>2</sub>	Q <sub>2</sub>
1	N <sub>1</sub>	A <sub>1</sub>	N <sub>2</sub>	4	P <sub>4</sub>	Q <sub>4</sub>
2	N <sub>1</sub>	A <sub>2</sub>	N <sub>3</sub>	2	P <sub>2</sub>	Q <sub>2</sub>
3	N <sub>2</sub>	A <sub>3</sub>	N <sub>1</sub>	1	P <sub>1</sub>	Q <sub>1</sub>
4	N <sub>3</sub>	A <sub>3</sub>	NULL	NULL	NULL	NULL
	NULL	NULL	NULL	3	P <sub>3</sub>	Q <sub>3</sub>

COLUMN STORE

# OLTP & OLAP access pattern

CUSTOMER			BUY			PRODUCT		
<u>CID</u>	CNAME	ADDRESS	<u>CID</u>	<u>PID</u>	AMOUNT	<u>PID</u>	PNAME	QUANTITY
1	N <sub>1</sub>	A <sub>1</sub>		2	N <sub>1</sub>		P <sub>1</sub>	Q <sub>1</sub>
2	N <sub>1</sub>	A <sub>2</sub>		4	N <sub>2</sub>	2	P <sub>2</sub>	Q <sub>2</sub>
	N <sub>2</sub>	A <sub>3</sub>	2	2	N <sub>3</sub>	3	P <sub>3</sub>	Q <sub>3</sub>
4	N <sub>3</sub>	A <sub>3</sub>		1	N <sub>4</sub>	4	P <sub>4</sub>	Q <sub>4</sub>

Q<sub>OLTP</sub> & Q<sub>OLAP</sub>

# OLTP & OLAP with WIDGETABLES

# CHALLENGES

CID	CNAME	ADDRESS	AMOUNT	PID	PNAME	QUANTITY
1	N <sub>1</sub>					
1	N <sub>1</sub>					
2	N <sub>2</sub>					
			NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

SCANNING FOR POINT  
ACCESS IS UNSMART

COLUMNS ARE CACHE IN-  
EFFICIENT FOR OLTP

NUMBER OF FIELDS PER  
COLUMN RESTRICT USAGE  
OF CO-PROC (OOM).

DATA REDUNDANCY +  
INTEGRITY ISSUES FOR OLTP

A LOT OF NULLS WASTE  
MEMORY IN ROW-STORES

UNIFIED QUERIES

INDEXES FOR FAST POINT ACCESS

ADAPTIVE PHYSICAL LAYOUT

UTILIZATION OF CO-PROCESSORS

SOLVE DE-NORMALIZE ISSUES

TABLE SPARSITY (NULLs)