C-Store: A Column Oriented DBMS

Michael	Stonebraker	MIT		
Daniel	Abadi	MIT		
Adam	Batkin	Brandeis		
Xuedong	Chen	UMass		
Mitch	Cherniack	Brandeis		
Miguel	Ferreira	MIT		
Edmond	Lau	MIT		
Amerson	Lin	MIT		
Sam	Madden	MIT		
Elizabeth	O'Neil	UMass		
Pat	O'Neil	UMass		
Alex	Rasin	Brown		
Nga	Tran	Brandeis		
Stan	Zdonik	Brown		

Q1. *Determine the total number of lineitems shipped for each day after day D.*

```
SELECT l_shipdate, COUNT (*)
FROM lineitem
WHERE l_shipdate > D
GROUP BY l_shipdate
```

			164x	21x

Query	C-Store	Row Store	Column Store
Q1	0.03	6.80	2.24
Q2	0.36	1.09	0.83
Q3	4.90	93.26	29.54
Q4	2.09	722.90	22.23
Q5	0.31	116.56	0.93
Q6	8.50	652.90	32.83
Q7	2.54	265.80	33.24

C-Store	Row Store	Column Store
1.987 GB	4.480 GB	2.650 GB

Row-oriented storage

Name	Age	Dept	Year
Andrew	24	CS	G1
Fred	28	Chemistry	G3
Vanessa	25	History	G1

...

```
SELECT name
FROM students
WHERE dept == 'CS'
```

Andrew, 24, CS, G1	Fred, 28, Chemistry, G3	Vanessa, 25, History, G1
Matthew, 30, Economics, G1	Mark, 25, CS, G1	Ann, 27, Music, G5

Column-oriented storage

Name	Age	Dept	Year
Andrew	24	CS	G1
Fred	28	Chemistry	G3
Vanessa	25	History	G1

. . .

Andrew, Fred, Vanessa, Matthew, Mark, Ann
24, 28, 25, 30, 25, 27
CS, Chemistry, History, Economics, CS, Music
G1, G3, G1, G1, G1, G5

```
SELECT name FROM students WHERE dept == 'CS'
```

Column-oriented storage

Name	Age	Dept	Year
Andrew	24	CS	G1
Fred	28	Chemistry	G3
Vanessa	25	History	G1

. . .

Andrew, Fred, Vanessa, Matthew, Mark, Ann

24, 28, 25, 30, 25, 27

CS, Chemistry, History, Economics, CS, Music

G1, G3, G1, G1, G1, G5

SELECT name FROM students WHERE dept == 'CS'

Column-oriented storage

Name	Age	Dept	Year
Andrew	24	CS	G1
Fred	28	Chemistry	G3
Vanessa	25	History	G1

· · ·

→

Andrew, Mark, Vanessa, Ann, Fred, Matthew

24, 25, 25, 27, 28, 30   sorted by age

CS, Chemistry, History, Economics, CS, Music

G1, G3, G1, G1, G1, G5

```
SELECT name FROM students ORDER BY age
```
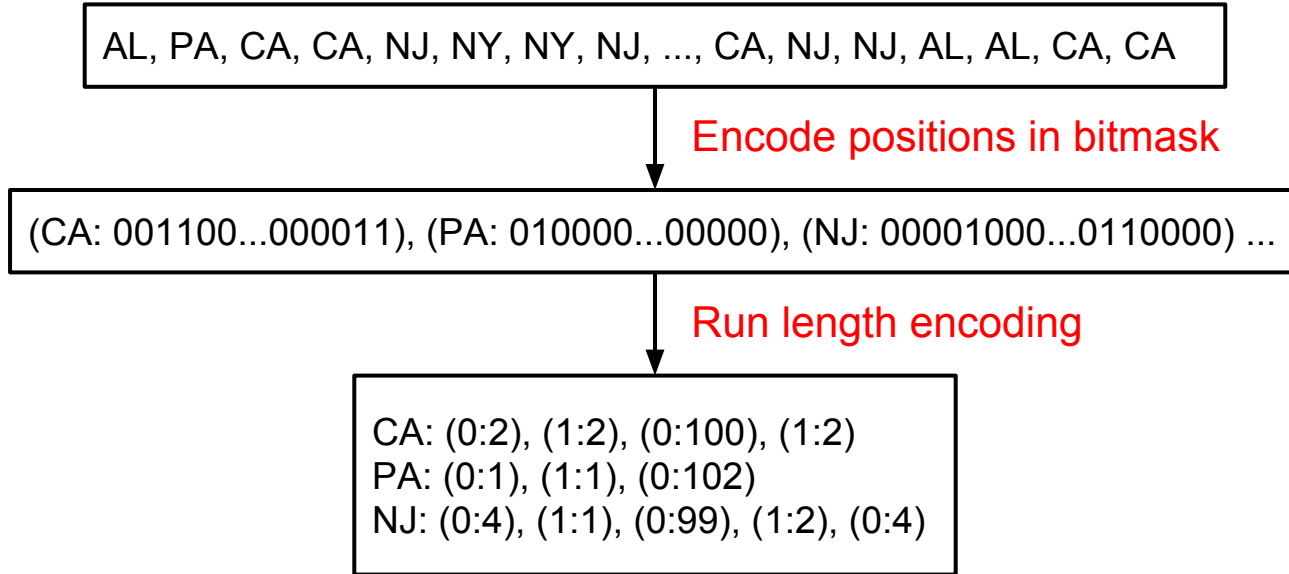
Column-oriented storage

22, 22, 23, 23, 23, 23, 24, 24, 24, …, 24, 24, 25, 25, 25, 25, 26, 26, 26, 26, 26, 26, 27, 27, 27

Run length encoding

(22: 2), (23: 4), (24: 16), (25: 4), (26: 6), (27: 3)

Column-oriented storage

AL, PA, CA, CA, NJ, NY, NY, NJ, ..., CA, NJ, NJ, AL, AL, CA, CA

Encode positions in bitmask

(CA: 001100...000011), (PA: 010000...00000), (NJ: 00001000...0110000) ...

Run length encoding

CA: (0:2), (1:2), (0:100), (1:2)
PA: (0:1), (1:1), (0:102)
NJ: (0:4), (1:1), (0:99), (1:2), (0:4)

Column-oriented storage

Name	Age	Dept	Year
Andrew	24	CS	G1
Fred	28	Chemistry	G3
Vanessa	25	History	G1

. . .

```
SELECT name
FROM students
ORDER BY age
```

Andrew, Mark, Vanessa, Ann, Fred, Matthew

24, 25, 25, 27, 28, 30   sorted by age

Andrew, Mark, Vanessa, Matthew, Fred, Ann

24, 25, 25, 27, 28, 30   sorted by year

Andrew, Ann, Fred, Mark, Matthew, Vanessa

CS, Music, Chemistry, CS, Economics, History

G1, G5, G3, G1, G1, G1   sorted by name

Columnar storage key ideas

Avoid reading columns not used in query

More efficient storage due to compression

Multiple column ordering due to reduced space

Operate directly on compressed data
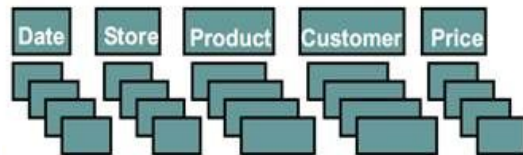
Row-oriented

Write-optimized, great for OLTP

Simple, battle tested by traditional DBMS

Reads must read in all columns and incur lots of random seeks
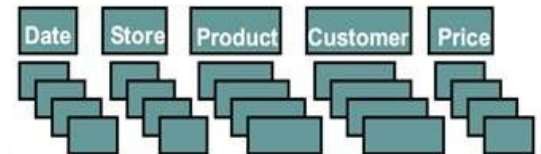
High storage cost

Column-oriented

Read-optimized, great for OLAP

Must maintain join indices to reconstruct complete rows

Reads can skip unused columns and often incur sequential scans

Low storage cost

C-Store

Read-optimized columnar database

Support many column sort-orders through *projections*

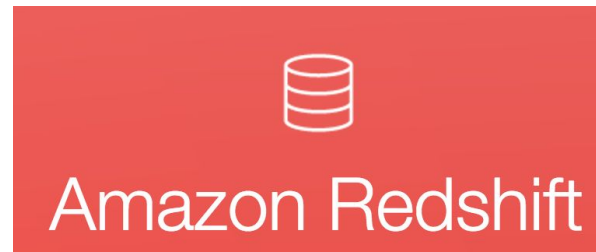Writeable Store to handle inserts and updates efficiently

Distributed transactions using 2PC

Snapshot isolation for read availability

VERTICA

druid

InfiniDB

monetdb

Amazon Redshift

Parquet

Apache ORC