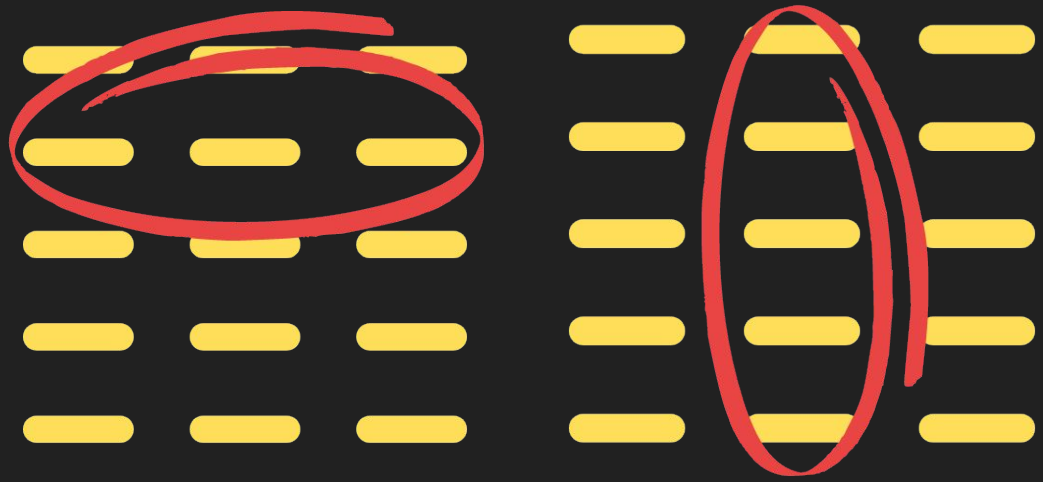


Row vs Column Oriented Databases

Database Storage

husseinnasser.com

Agenda

- Row-Oriented Database (Row store)
- Column-Oriented Database (Column store)
- Pros & Cons

Table

rowid	id	first_name	last_name	ssn	salary	dob	title	joined
1001	1	John	Smith	111	101,000	1/1/1991	eng	1/1/2011
1002	2	Kary	White	222	102,000	2/2/1992	mgr	2/1/2012
1003	3	Norman	Freeman	333	103,000	3/3/1993	mkt	3/1/2013
1004	4	Nole	Smith	444	104,000	4/4/1994	adm	4/1/2014
1005	5	Dar	Sol	555	105,000	5/5/1995	adm	5/1/2015
1006	6	Yan	Thee	666	106,000	6/6/1996	mkt	6/1/2016
1007	7	Hasan	Ali	777	107,000	7/7/1997	acc	7/1/2017
1008	8	Ali	Bilal	888	108,000	8/8/1998	acc	8/1/2018

Queries

- No indexes
- Select first_name from emp where ssn = 666
- Select * from emp where id = 1
- Select sum(salary) from emp

Row-Oriented Database

- Tables are stored as rows in disk
- A single block io read to the table fetches multiple rows with all their columns.
- More IOs are required to find a particular row in a table scan but once you find the row you get all columns for that row.

Row-Oriented Database

rowid	id	first_name	last_name	ssn	salary	dob	title	joined
1001	1	John	Smith	111	101,000	1/1/1991	eng	1/1/2011
1002	2	Kary	White	222	102,000	2/2/1992	mgr	2/1/2012
1003	3	Norman	Freeman	333	103,000	3/3/1993	mkt	3/1/2013
1004	4	Nole	Smith	444	104,000	4/4/1994	adm	4/1/2014
1005	5	Dar	Sol	555	105,000	5/5/1995	adm	5/1/2015
1006	6	Yan	Thee	666	106,000	6/6/1996	mkt	6/1/2016
1007	7	Hasan	Ali	777	107,000	7/7/1997	acc	7/1/2017
1008	8	Ali	Bilal	888	108,000	8/8/1998	acc	8/1/2018

Row-Oriented Database

1001, 1, John, Smith, 111, 101,000, 1/1/1991, eng, 1/1/2011|||
1002,2,Kary,White,222,102,000,2/2/1992,mgr,2/1/2012

1003,3,Norman,Freeman,333,103,000,3/3/1993,mkt,3/1/2013|||
1004,4,Nole,Smith,444,104,000,4/4/1994,adm,4/1/2014

1005,5,Dar,Sol,555,105,000,5/5/1995,adm,5/1/2015|||
1006,6,Yan,Thee,666,106,000,6/6/1996,mkt,6/1/2016

1007,7,Hasan,Ali,777,107,000,7/7/1997,acc,7/1/2017|||
1008,8,Ali,Bilal,888,108,000,8/8/1998,acc,8/1/2018

Select first_name from emp where ssn=666

1001, 1, John, Smith, 111, 101,000, 1/1/1991, eng, 1/1/2011|||
1002,2,Kary,White,222,102,000,2/2/1992,mgr,2/1/2012

1003,3,Norman,Freeman,333,103,000,3/3/1993,mkt,3/1/2013|||
1004,4,Nole,Smith,444,104,000,4/4/1994,adm,4/1/2014

1005,5,Dar,Sol,555,105,000,5/5/1995,adm,5/1/2015|||
1006,6,Yan,Thee,666,106,000,6/6/1996,mkt,6/1/2016

Select * from Emp where id = 1

1001, 1, John, Smith, 111, 101,000, 1/1/1991, eng, 1/1/2011|||
1002,2,Kary,White,222,102,000,2/2/1992,mgr,2/1/2012

Select sum(salary) from emp

1001, 1, John, Smith, 111, 101,000, 1/1/1991, eng, 1/1/2011|||
1002,2,Kary,White,222,102,000,2/2/1992,mgr,2/1/2012

1003,3,Norman,Freeman,333,103,000,3/3/1993,mkt,3/1/2013|||
1004,4,Nole,Smith,444,104,000,4/4/1994,adm,4/1/2014

1005,5,Dar,Sol,555,105,000,5/5/1995,adm,5/1/2015|||
1006,6,Yan,Thee,666,106,000,6/6/1996,mkt,6/1/2016

1007,7,Hasan,Ali,777,107,000,7/7/1997,acc,7/1/2017|||
1008,8,Ali,Bilal,888,108,000,8/8/1998,acc,8/1/2018

Column-Oriented Database

- Tables are stored as columns first in disk
- A single block io read to the table fetches multiple columns with all matching rows
- Less IOs are required to get more values of a given column. But working with multiple columns require more IOs.
- OLAP

Column-Oriented Database

rowid	id	first_name	last_name	ssn	salary	dob	title	joined
1001	1	John	Smith	111	101,000	1/1/1991	eng	1/1/2011
1002	2	Kary	White	222	102,000	2/2/1992	mgr	2/1/2012
1003	3	Norman	Freeman	333	103,000	3/3/1993	mkt	3/1/2013
1004	4	Nole	Smith	444	104,000	4/4/1994	adm	4/1/2014
1005	5	Dar	Sol	555	105,000	5/5/1995	adm	5/1/2015
1006	6	Yan	Thee	666	106,000	6/6/1996	mkt	6/1/2016
1007	7	Hasan	Ali	777	107,000	7/7/1997	acc	7/1/2017
1008	8	Ali	Bilal	888	108,000	8/8/1998	acc	8/1/2018

Column-Oriented Database

1:1001, 2:1002, 3:1003, 4:1004, 5:1005, 6:1006, 7:1007, 8:1008

John:1001, Kary:1002, Norman:1003, Nole:1004, Dar:1005, Yan:1006, Hasan:1007, Ali:1008

Smith:1001, White:1002, Freeman:1003, Sol:1004 Thee:1005, Ali:1006, Bilal:1007, Ali:1008

111:1001, 222:1002, 333:1003, 444:1004, 555:1005, 666:1006, 777:1007, 888:1008

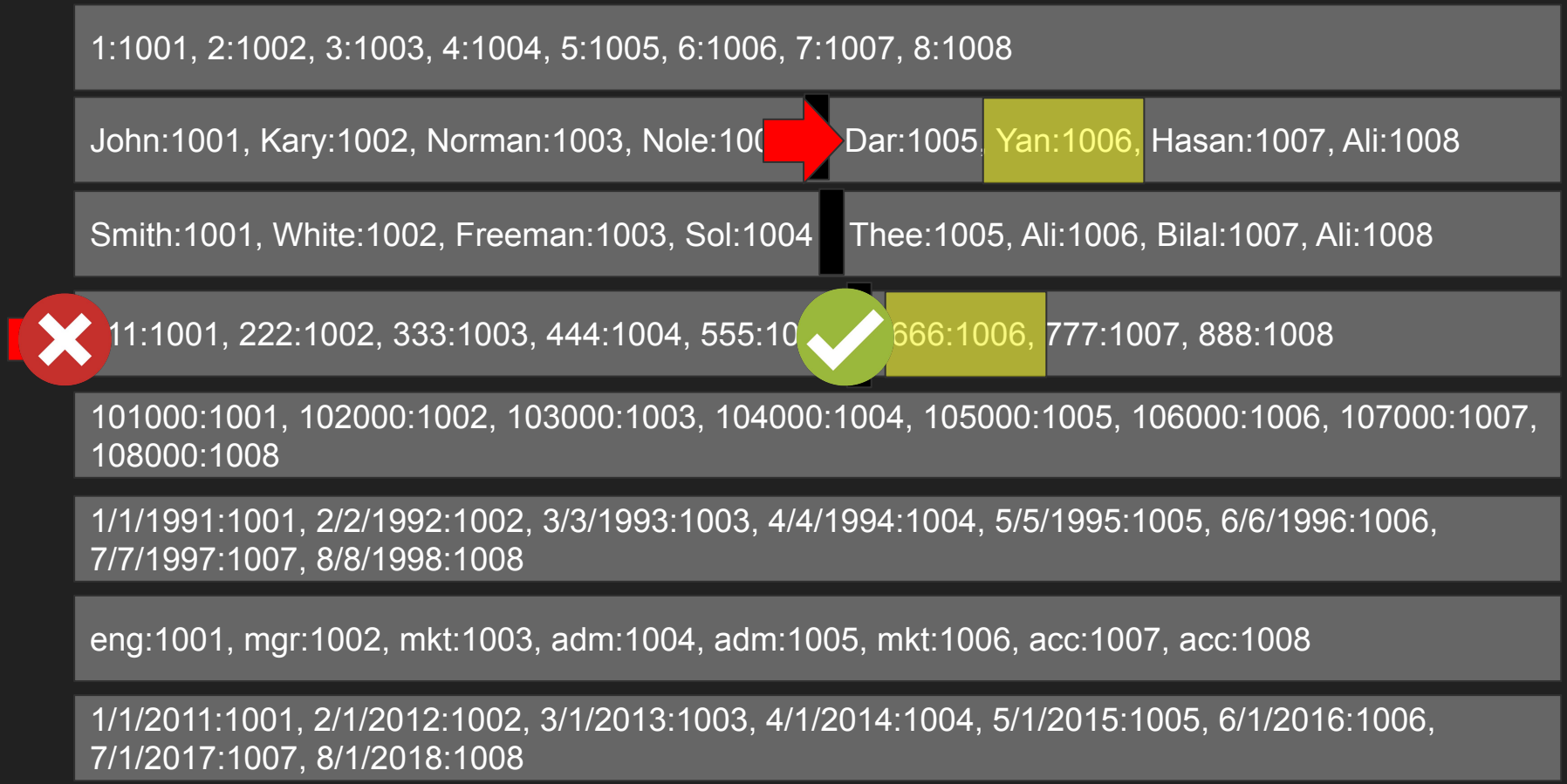
101000:1001, 102000:1002, 103000:1003, 104000:1004, 105000:1005, 106000:1006, 107000:1007, 108000:1008

1/1/1991:1001, 2/2/1992:1002, 3/3/1993:1003, 4/4/1994:1004, 5/5/1995:1005, 6/6/1996:1006, 7/7/1997:1007, 8/8/1998:1008

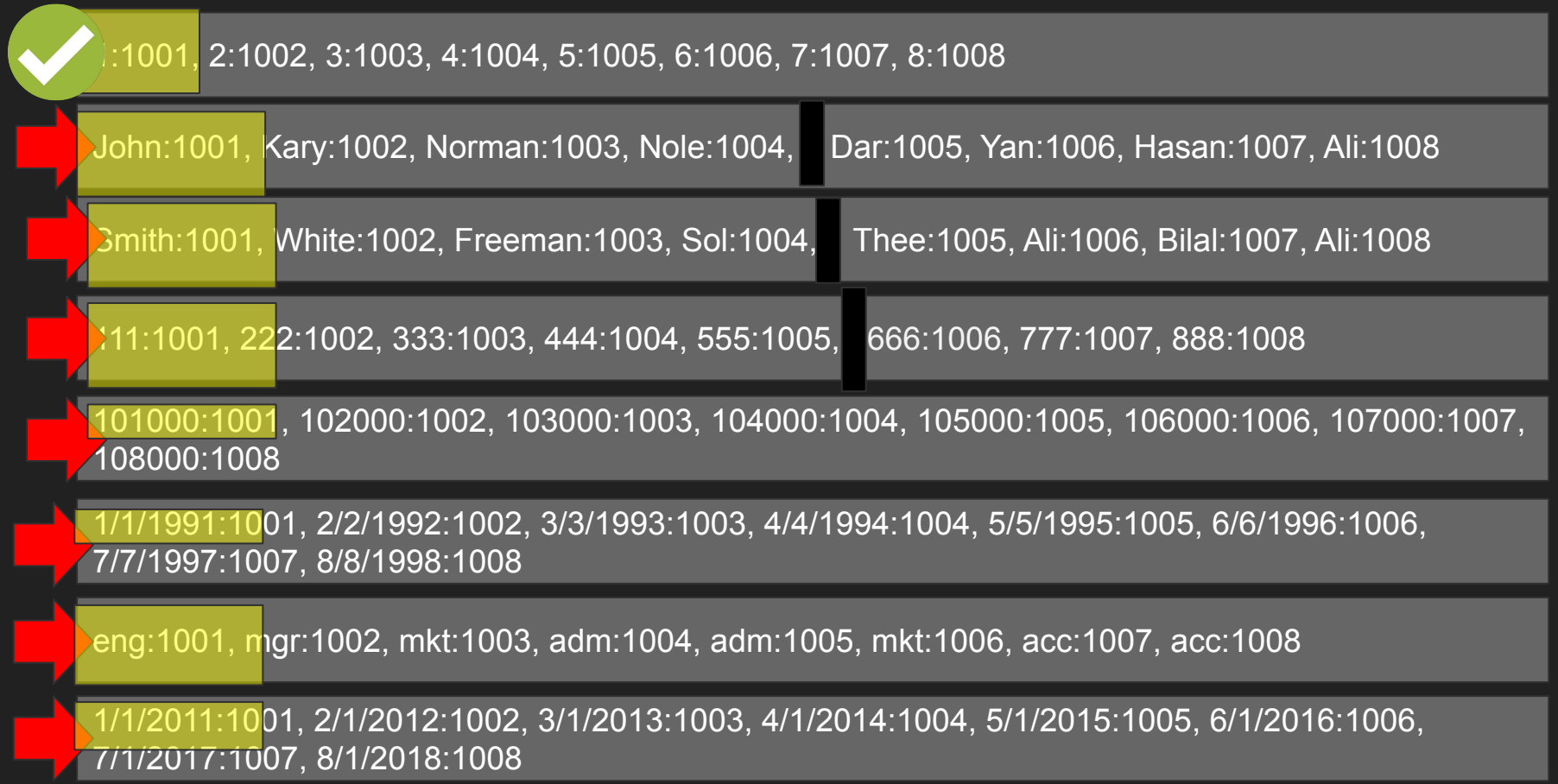
eng:1001, mgr:1002, mkt:1003, adm:1004, adm:1005, mkt:1006, acc:1007, acc:1008

1/1/2011:1001, 2/1/2012:1002, 3/1/2013:1003, 4/1/2014:1004, 5/1/2015:1005, 6/1/2016:1006, 7/1/2017:1007, 8/1/2018:1008

Select first_name from emp where ssn = 666



Select * from emp where id = 1



Select sum(salary) from emp

1:1001, 2:1002, 3:1003, 4:1004, 5:1005, 6:1006, 7:1007, 8:1008

John:1001, Kary:1002, Norman:1003, Nole:1004, Dar:1005, Yan:1006, Hasan:1007, Ali:1008

Smith:1001, White:1002, Freeman:1003, Sol:1004, Thee:1005, Ali:1006, Bilal:1007, Ali:1008

111:1001, 222:1002, 333:1003, 444:1004, 555:1005, 666:1006, 777:1007, 888:1008

101000:1001, 102000:1002, 103000:1003, 104000:1004, 105000:1005, 106000:1006, 107000:1007, 108000:1008

1/1/1991:1001, 2/2/1992:1002, 3/3/1993:1003, 4/4/1994:1004, 5/5/1995:1005, 6/6/1996:1006, 7/7/1997:1007, 8/8/1998:1008

eng:1001, mgr:1002, mkt:1003, adm:1004, adm:1005, mkt:1006, acc:1007, acc:1008

1/1/2011:1001, 2/1/2012:1002, 3/1/2013:1003, 4/1/2014:1004, 5/1/2015:1005, 6/1/2016:1006, 7/1/2017:1007, 8/1/2018:1008

Pros & Cons

- Row-Based
- Optimal for read/writes
- OLTP
- Compression isn’t efficient
- Aggregation isn’t efficient
- Efficient queries w/multi-columns

- Column-Based
- Writes are slower
- OLAP
- Compress greatly
- Amazing for aggregation
- Inefficient queries w/multi-columns