



# Andreas Jordan

## Bufferpool-Fragmentierung beim SQL Server erkennen

#ittage

# Bufferpool-Fragmentierung beim SQL Server erkennen

Andreas Jordan

IT-Tage

14.12.2023, Frankfurt am Main



# Andreas Jordan

ORDIX AG

- Principal Consultant und Teamleiter
- Microsoft SQL Server
- Oracle Database
- PowerShell
- Sprecher auf den IT-Tagen 2015, 2022, 2023
- Sprecher auf der DOAG Konferenz & Ausstellung 2022, 2023
- <https://blog.ordix.de/andreas-jordan>



# Agenda

ORDIX AG

- Fragmentierung: Egal oder schädlich?
- Welche Seiten sind im Hauptspeicher?
- Fragmentierung ganz klassisch ermitteln
- Meine Empfehlungen



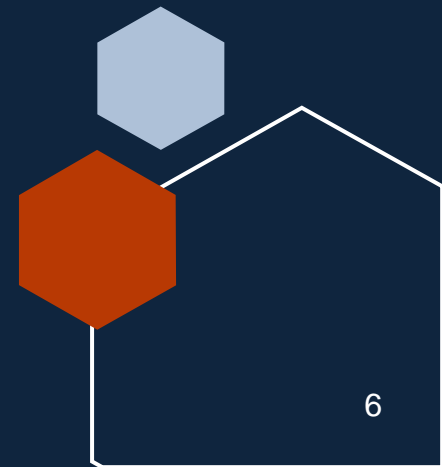
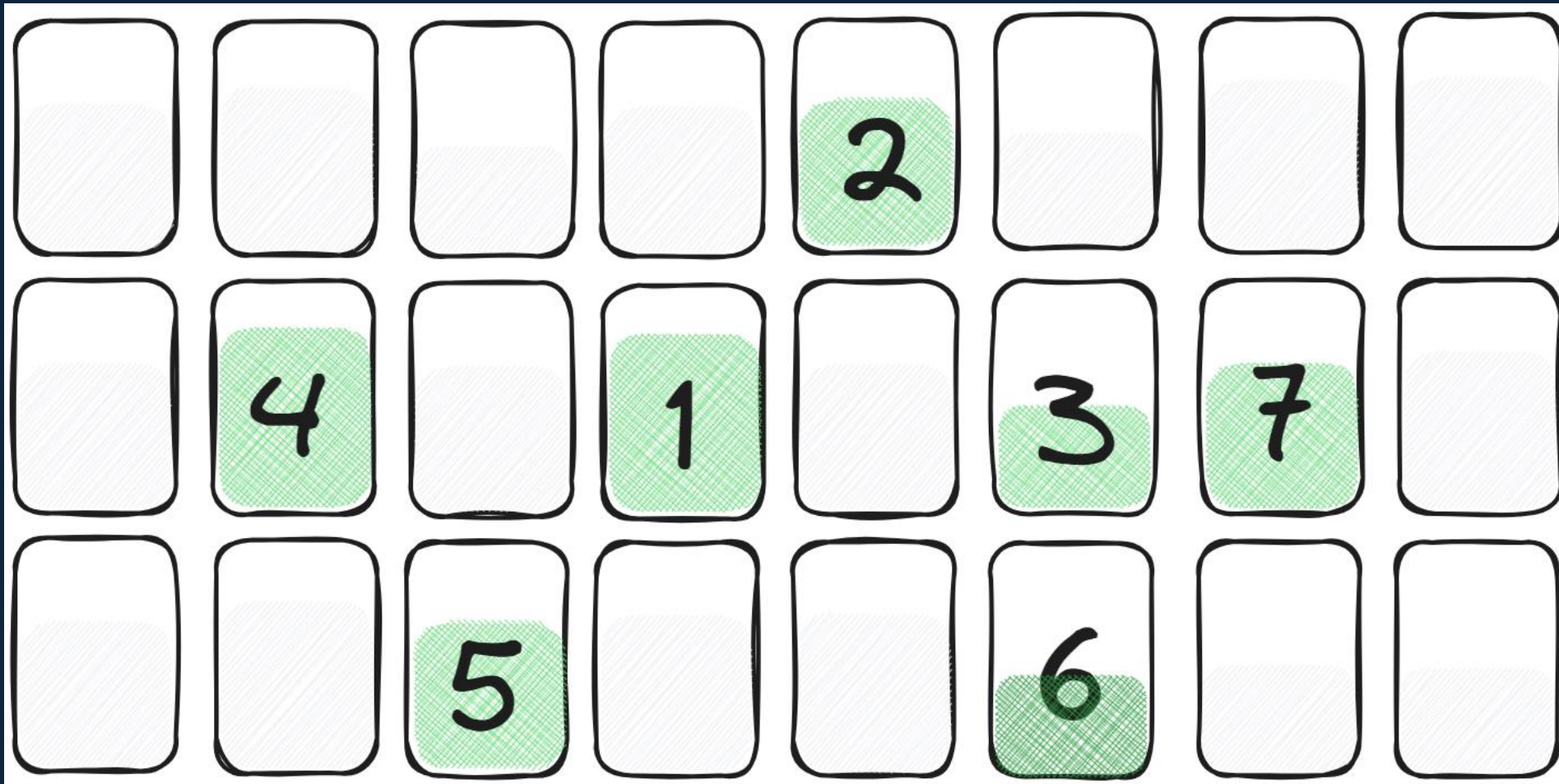
# Fragmentierung: Egal oder schädlich?

- Vorab:
  - Die Datenbank teilt alle Daten in 8 kByte große Seiten (pages) ein.
  - Dies ist die kleinste Einheit, die zwischen der Festplatte und dem Hauptspeicher ausgetauscht wird.
- Externe Fragmentierung:
  - Die Position der Seiten innerhalb der Datendatei ist nicht optimal.
- Interne Fragmentierung:
  - Der Füllgrad der Seiten ist nicht optimal.
- Heute im Fokus:
  - Die Seiten im Hauptspeicher.



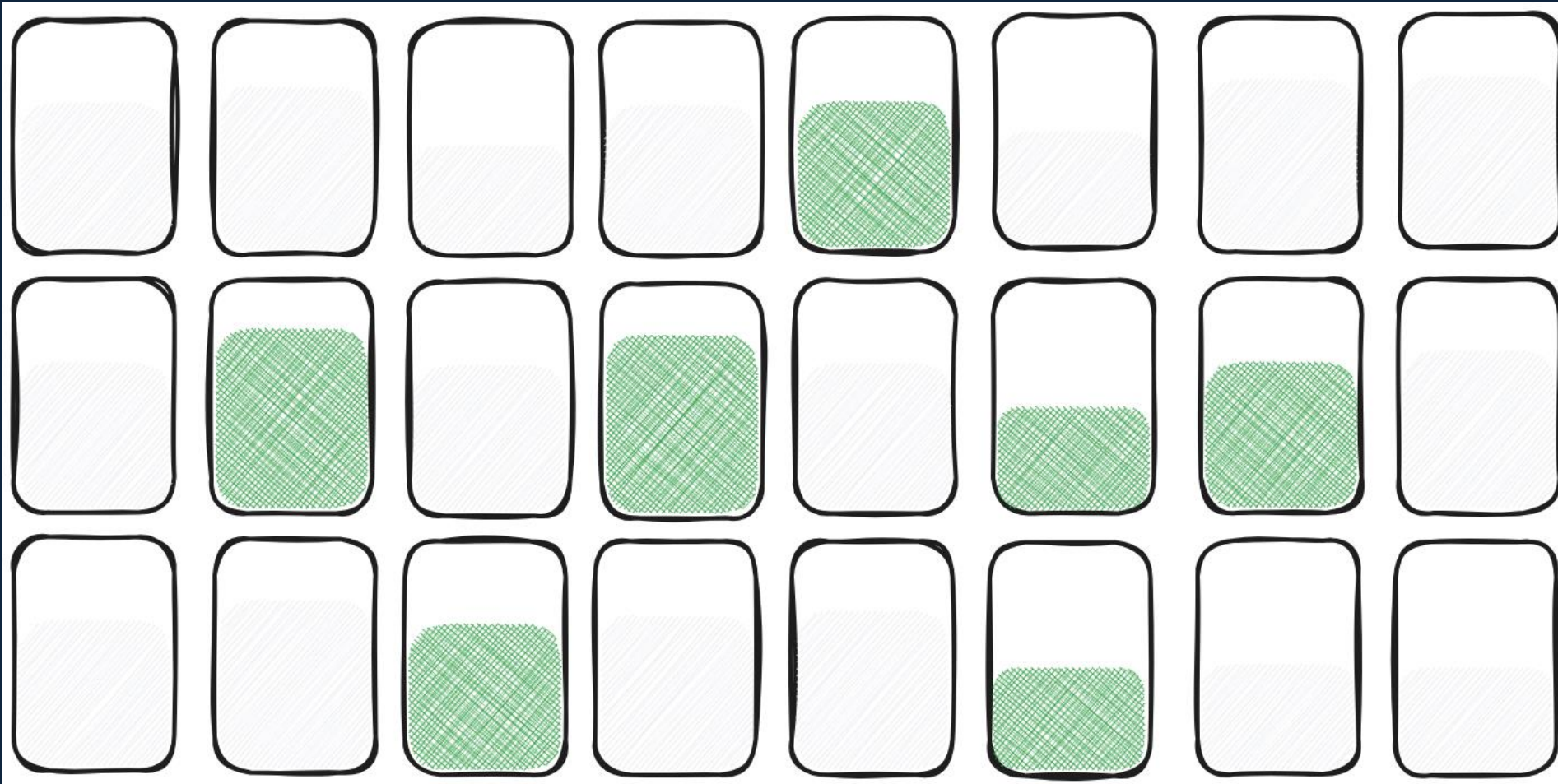
# Fragmentierung: Egal oder schädlich?

ORDIX AG



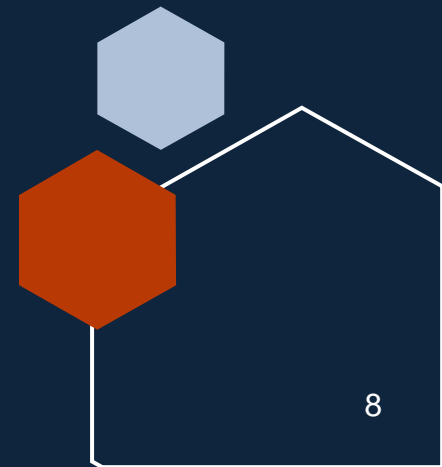
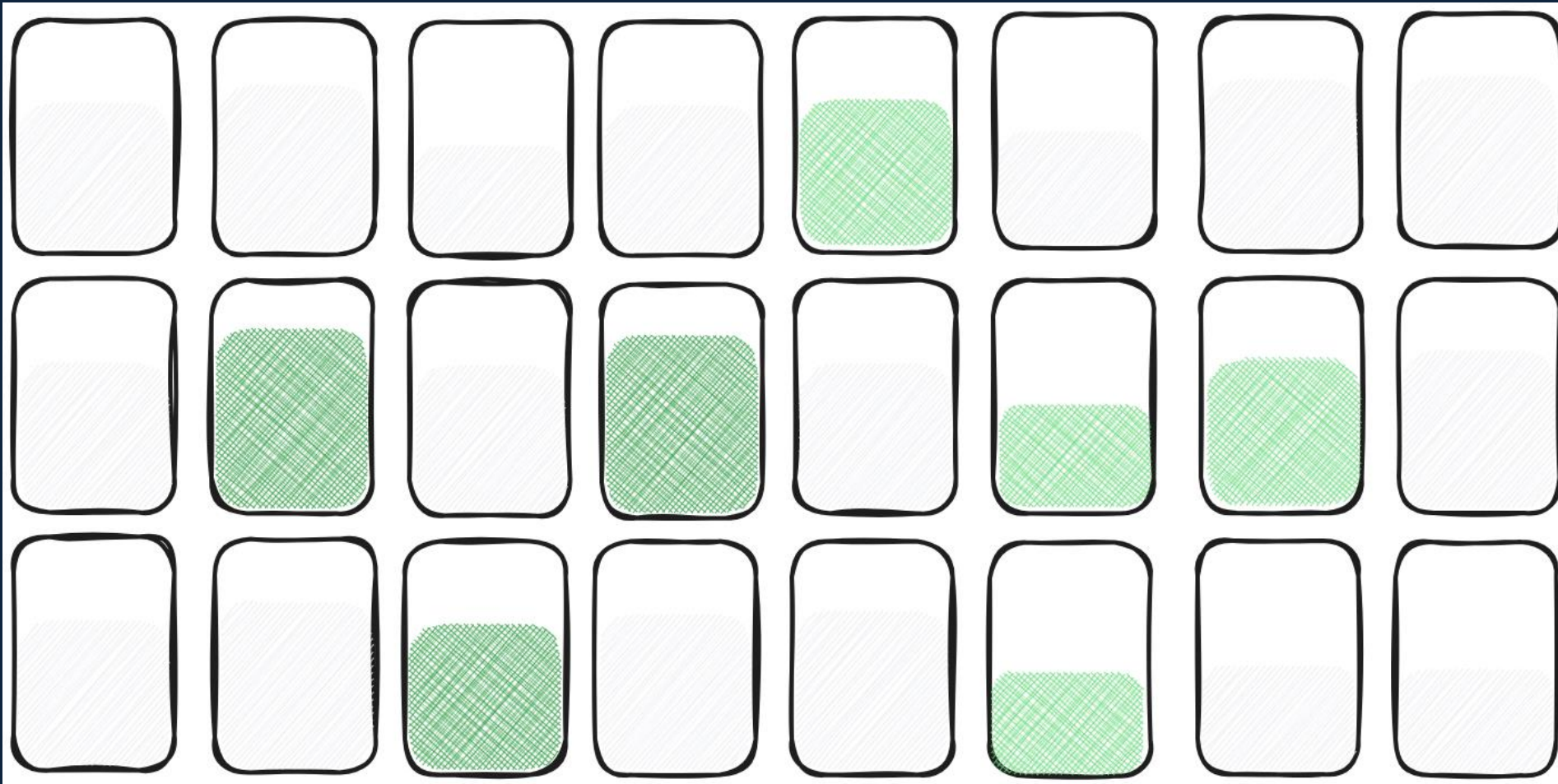


# Fragmentierung: Egal oder schädlich?



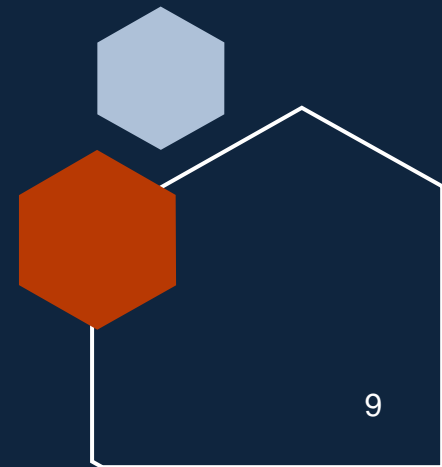
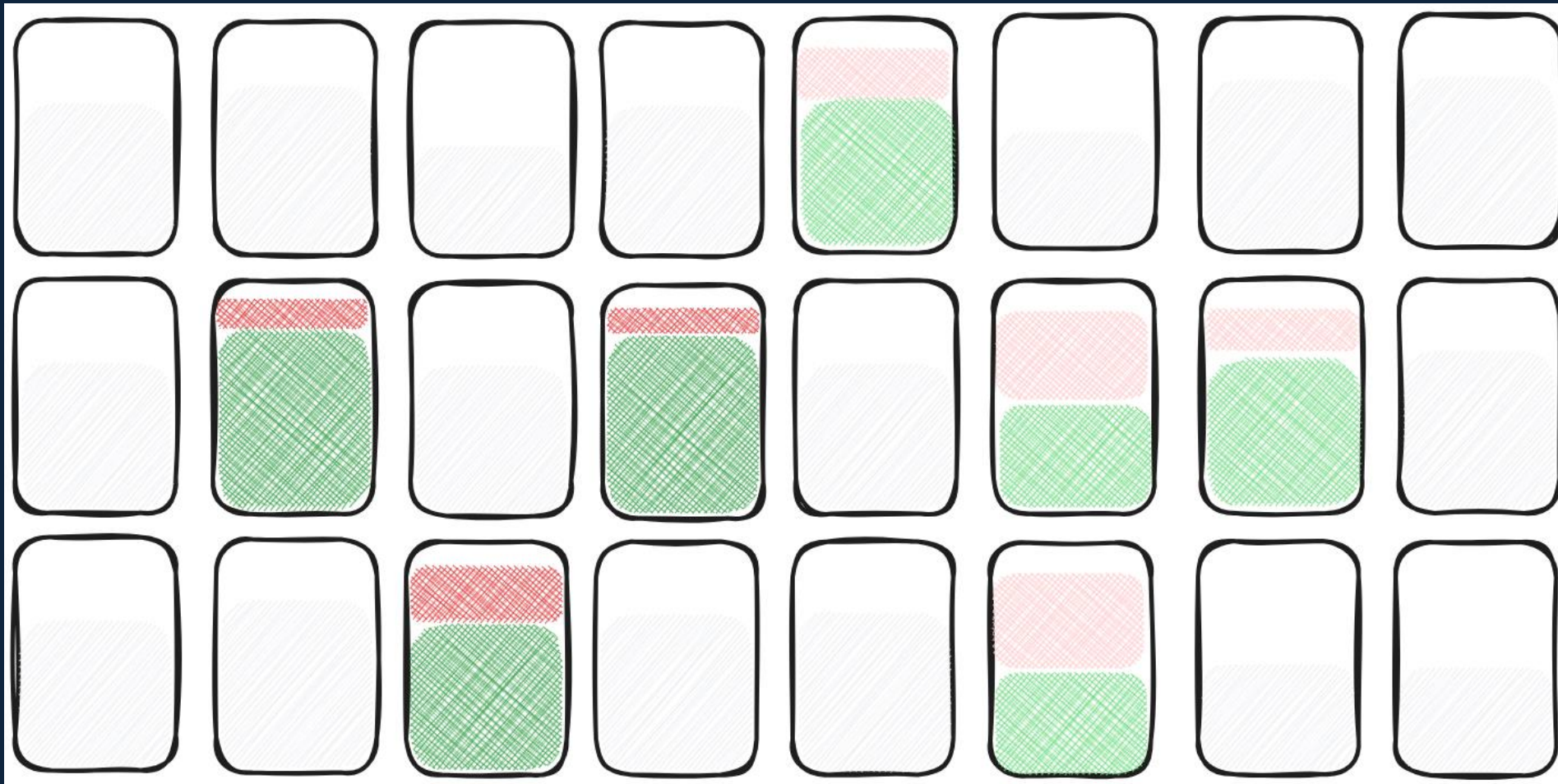
# Fragmentierung: Egal oder schädlich?

ORDIX AG





# Fragmentierung: Egal oder schädlich?



# Welche Seiten sind im Hauptspeicher?

- Die Antwort darauf hat die Systemsicht **sys.dm\_os\_buffer\_descriptors**.

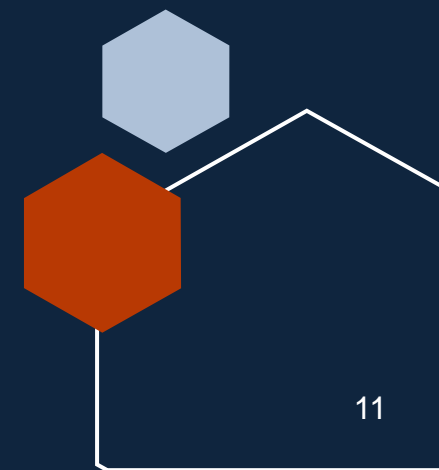
Column name	Data type	Description
database_id	int	ID of database associated with the page in the buffer pool.
file_id	int	ID of the file that stores the persisted image of the page.
page_id	int	ID of the page within the file.
page_level	int	Index level of the page.
allocation_unit_id	bigint	ID of the allocation unit of the page. This value can be used to join sys.allocation_units.
page_type	nvarchar(60)	Type of the page, such as: Data page or Index page.
row_count	int	Number of rows on the page.
free_space_in_bytes	int	Amount of available free space, in bytes, on the page.

# Welche Seiten sind im Hauptspeicher?

- Wie sieht das in der Praxis aus?

Cached Data MB	Free MB	% Free	DB Name
175.749,02	39.844,72	<div></div>	22,0 ***ALL DATABASES***
173.938,23	39.349,53	<div></div>	22,6 TCPRD
1.246,66	347,03	<div></div>	27,8 tempdb
341,95	128,73	<div></div>	37,7 TCRAP_13_DB
209,53	13,35	<div></div>	6,4 msdb
5,32	2,02	<div></div>	38,0 ResourceDb
3,55	2,22	<div></div>	62,5 TcClusterDB_TCPRD
2,52	1,17	<div></div>	46,4 master
1,24	0,57	<div></div>	46,0 model

Cached Data MB	Free MB	% Free	DB Name
25.765,27	10.310,18	<div></div>	40,0 ***ALL DATABASES***
25.543,92	10.285,73	<div></div>	40,3 tc
189,42	12,53	<div></div>	6,6 tempdb
16,31	6,72	<div></div>	41,2 ResourceDb
10,70	3,14	<div></div>	29,4 msdb
1,99	0,89	<div></div>	44,7 master
1,83	0,66	<div></div>	36,1 TcClusterDB
1,08	0,44	<div></div>	40,7 model



# Welche Seiten sind im Hauptspeicher?

- Wie sieht das in der Praxis aus?

Cached Data MB	Free MB	% Free	DB Name
295,32	32,25	<div></div>	9,0 ***ALL DATABASES***
226,74	14,84	<div></div>	6,5 COPRA6Test
24,44	2,14	<div></div>	8,8 COPRA6Design
20,38	7,02	<div></div>	34,5 ResourceDb
12,65	3,60	<div></div>	28,5 tempdb
7,27	2,75	<div></div>	37,8 msdb
2,27	1,15	<div></div>	50,7 master
1,57	0,75	<div></div>	47,8 model

# Welche Seiten sind im Hauptspeicher?

- Geht das per SQL?

```
SELECT database_id
      , DB_NAME(database_id) AS database_name
      , COUNT(*) AS pages
      , COUNT(*)*8/1024 AS mb_total
      , SUM(CAST(free_space_in_bytes as bigint))/1024/1024 AS mb_free
FROM sys.dm_os_buffer_descriptors
WHERE database_id BETWEEN 5 and 32760
GROUP BY database_id
ORDER BY SUM(CAST(free_space_in_bytes as bigint)) DESC;
```

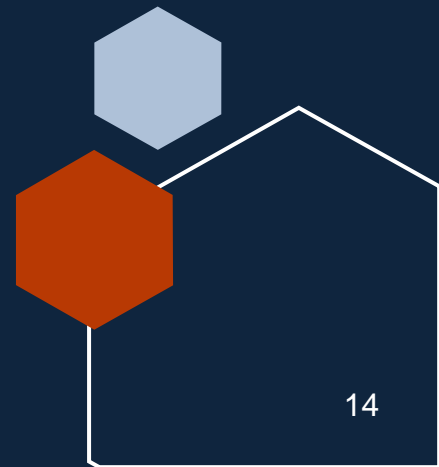
database_id	database_name	pages	mb_total	mb_free	pct_free
5		2207142	17243	4983	28
6	_PRODU	119556	934	305	32
7	_PRINT	80949	632	28	4
9	_T	14835	115	14	12
8	TRANSFER	438	3	1	35





# Welche Seiten sind im Hauptspeicher?

database_id	database_name	page_type_and_level	pages	mb_total	mb_free
5		DATA_PAGE	1967816	15373	4752
6	_PRODU	DATA_PAGE	146784	1146	281
5		INDEX_PAGE_non_leaf	25530	199	71
5		INDEX_PAGE_leaf	157891	1233	71
7	_PRINT	DATA_PAGE	5051	39	22
6	_PRODU	INDEX_PAGE_leaf	13061	102	19



# Welche Seiten sind im Hauptspeicher?

table_name	index_id	index_name	index_type	page_type_and_level	pages	mb_total	mb_free	avg_free_space_in_bytes	avg_page_space_used_in_percent
EKPOS	1	PK_EKPOS	CLUSTERED	DATA_PAGE	761544	5949	1704	2346	71
AUFARTIK	1	PK_AUFARTIK	CLUSTERED	DATA_PAGE	480760	3755	1324	2887	64
AUFKOPF	1	PK_AUFKOPF	CLUSTERED	DATA_PAGE	125746	982	297	2479	69
EKKOPF	1	PK_EKKOPF	CLUSTERED	DATA_PAGE	131163	1024	222	1782	77
GRP	1	PK_GRP	CLUSTERED	DATA_PAGE	86046	672	215	2621	67
LIEFPOS	1	PK_LIEFPOS	CLUSTERED	DATA_PAGE	52171	407	179	3617	55
AUFTEXT	1	PK_AUFTEXT	CLUSTERED	DATA_PAGE	54944	429	170	3250	59
LIEFKOPF	1	PK_LIEFKOPF	CLUSTERED	DATA_PAGE	24500	191	70	3000	62
EKLIEFK	1	PK_EKLIEFK	CLUSTERED	DATA_PAGE	18780	146	60	3379	58
MWBUCH_H	1	PK_MWBUCH_H	CLUSTERED	DATA_PAGE	18932	147	43	2385	70
GESTELL_HIST	1	PK_GESTELL_HIST	CLUSTERED	DATA_PAGE	15281	119	42	2889	64
BE_KOPF	1	PK_BE_KOPF	CLUSTERED	DATA_PAGE	13789	107	41	3160	60
BELEGARCHIV	1	PK_BELEGARCHIV	CLUSTERED	DATA_PAGE	18639	145	39	2222	72
AUFPREIS	1	PK_AUFPREIS	CLUSTERED	DATA_PAGE	12612	98	38	3188	60
RECHKOPF	1	PK_RECHKOPF	CLUSTERED	DATA_PAGE	38494	300	33	919	88
AUFPOS	1	PK_AUFPOS	CLUSTERED	DATA_PAGE	10851	84	29	2821	65

# Welche Seiten sind im Hauptspeicher?

- Welche Datenbanken sind dominant?
- Wie verändern sich die Anteile über die Zeit?
- Welche Objekte sind dominant?
- Wie verändern sich die Anteile über die Zeit?
- Wie verändert sich der freie Platz über die Zeit?



# Fragmentierung ganz klassisch ermitteln

- `sys.dm_db_index_physical_stats`
  - `avg_fragmentation_in_percent`
    - Berücksichtigt den kompletten Index.
    - Berücksichtigt vor allem externe Fragmentierung.
    - Wird von Wartungsplänen und Ola Hallengren genutzt.
  - `avg_page_space_used_in_percent`
    - Berücksichtigt nur die interne Fragmentierung.
    - Die Seiten müssen in den Hauptspeicher geladen werden.
    - Ist nur bei SAMPLED und DETAILED vorhanden, nicht bei LIMITED.



- Regelmäßige Defragmentierung mit der Maintenance Solution von Ola Hallengren.
  - Schwellwerte anpassen.
  - LogToTable verwenden.

➔ So wenig Defragmentierung wie möglich, so viel wie nötig.

- Regelmäßige Analyse der Hauptspeicherverwendung mit `sys.dm_os_buffer_descriptors`.
  - Wie viel “Luft” befindet sich im Hauptspeicher?
  - Welche Indizes sind dafür verantwortlich?

➔ Bei wenig “Luft” muss nicht defragmentiert werden.





A decorative graphic on the left side of the slide. It features a large orange hexagon in the center. To its top right is a light blue hexagon. To its bottom left is a white outline of a hexagon. Below the large orange hexagon is a smaller orange hexagon.

**Vielen Dank für  
Ihre Aufmerksamkeit**

# **ORDIX AG**

Aktiengesellschaft für  
Softwareentwicklung,  
Schulung, Beratung und  
Systemintegration

Zentrale Paderborn  
Karl-Schurz-Straße 19a  
33100 Paderborn  
Tel.: 05251 1063-0  
Fax: 0180 1 67349 0

Seminarzentrum Wiesbaden  
Kreuzberger Ring 13  
65205 Wiesbaden  
Tel.: 0611 77840-00

info@ordix.de  
<https://www.ordix.de/>