

ORACLE®

Optimiertes Laden in die F-Fakten-Tabelle

SAP Business Warehouse

Jörn Bartels

Architect

Server Technologies – SAP Development

30. Juni 2014

Agenda

- 1 ➤ Standard Laden der F-Fakten Tabelle
- 2 ➤ Erste Optimierung
- 3 ➤ Unusable Indexes / Table Factoring
- 4 ➤ Optimierung mit Hinweis 1842044
- 5 ➤ Ausblick

Agenda

- 1 Standard Laden der F-Fakten Tabelle
- 2 Erste Optimierung
- 3 Unusable Indexes / Table Factoring
- 4 Optimierung mit Hinweis 1842044
- 5 Ausblick

Standard Laden der F-Fakten Tabelle

Laden gegen Indices

Laden gegen Indices

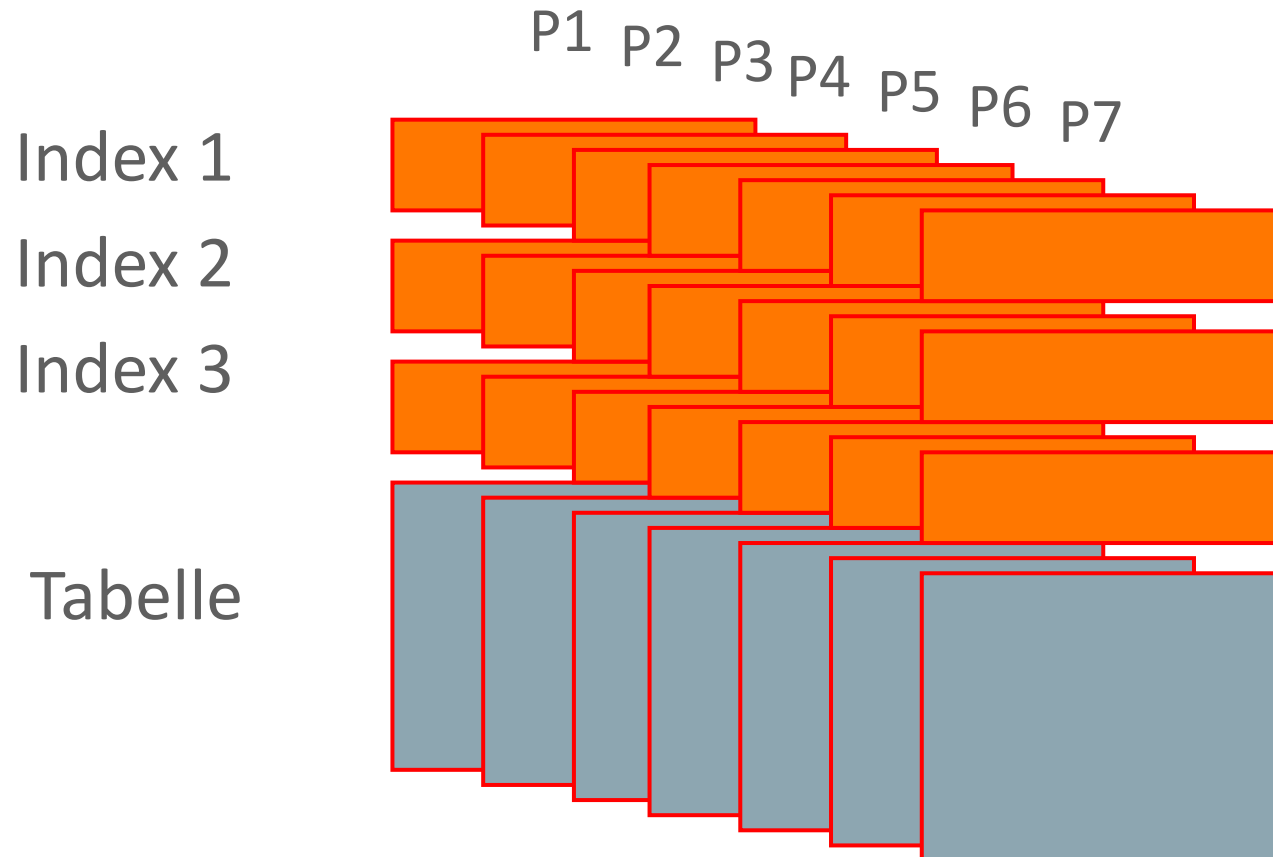
Beispiel

Tabelle: "/BI0/FOATR_CR01"

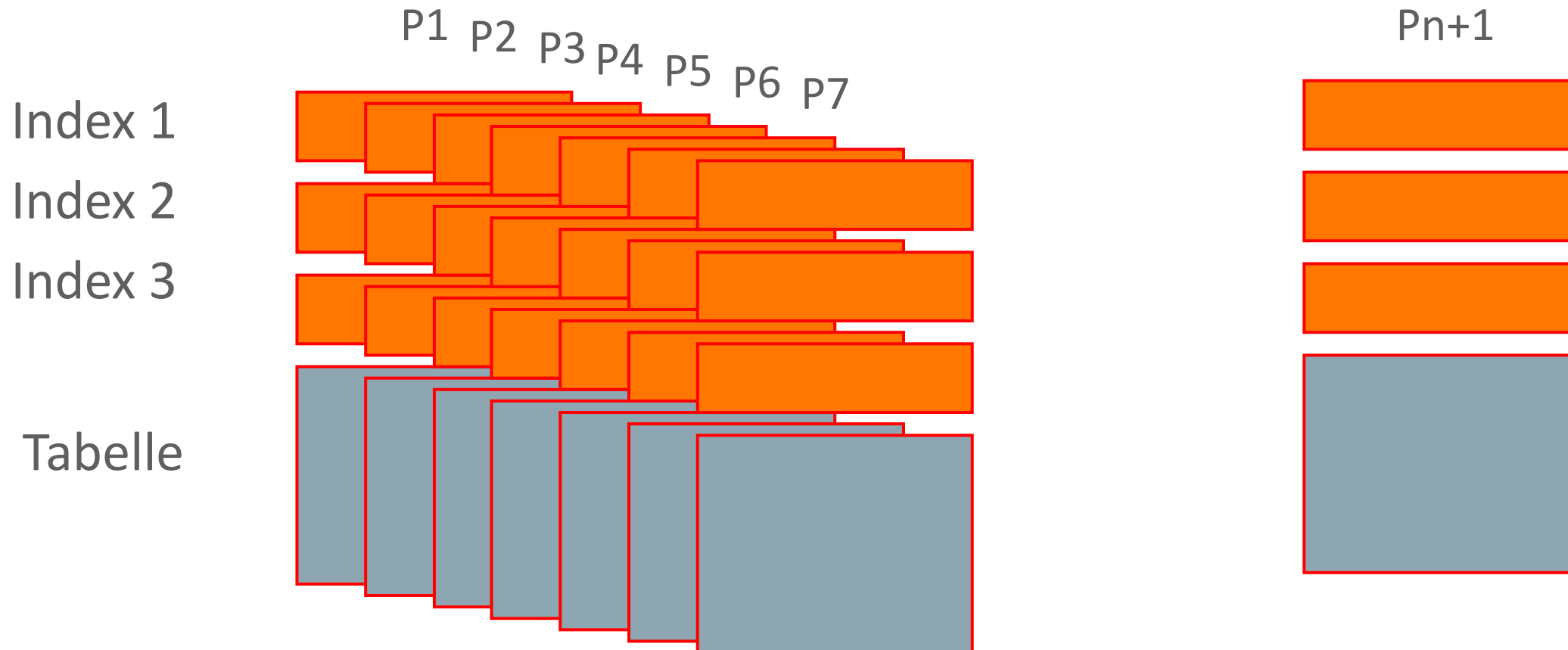
Zeilen: 16,5 Millionen

Partitionen: 68

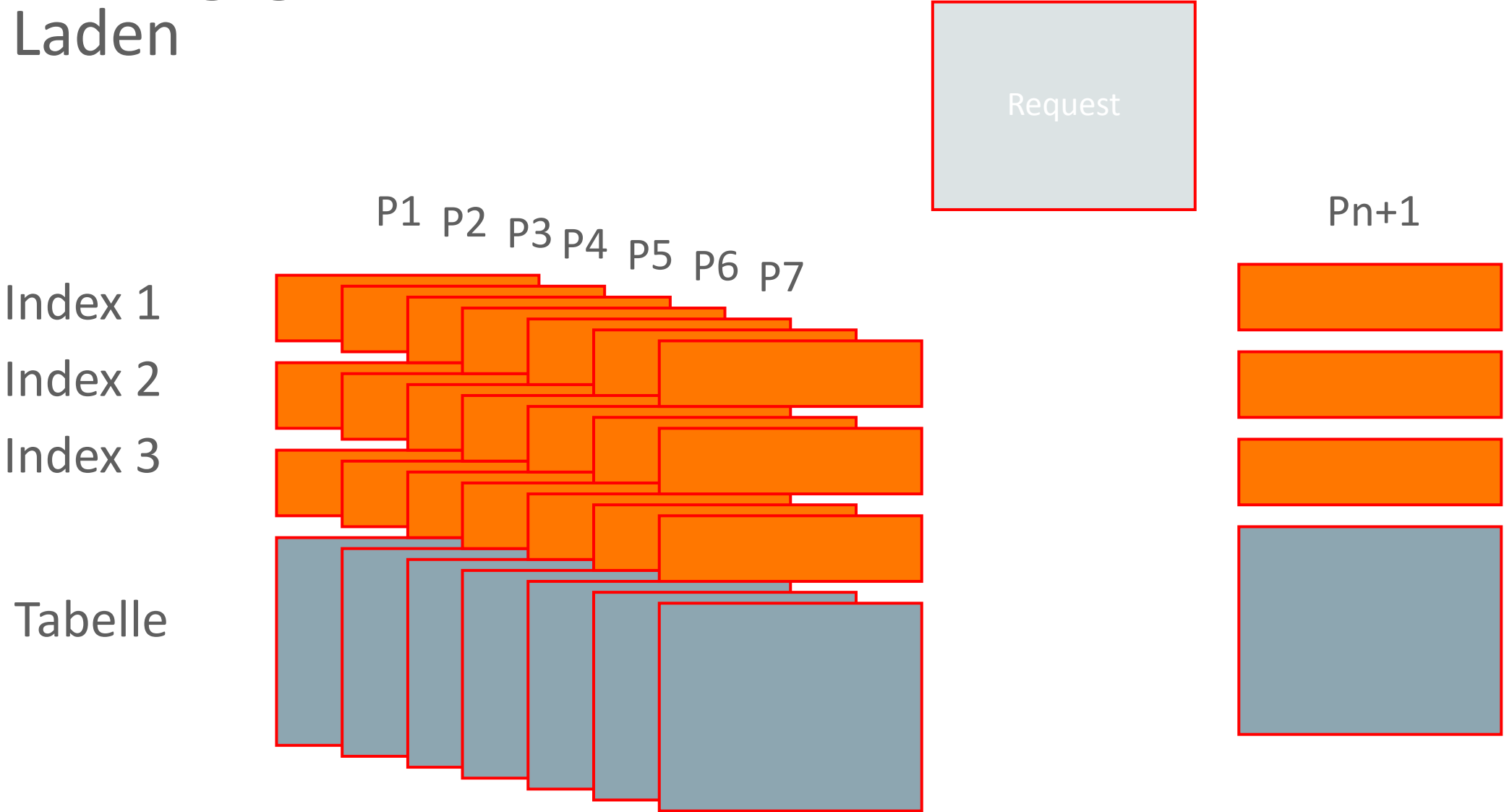
Größe: 1,2 GB



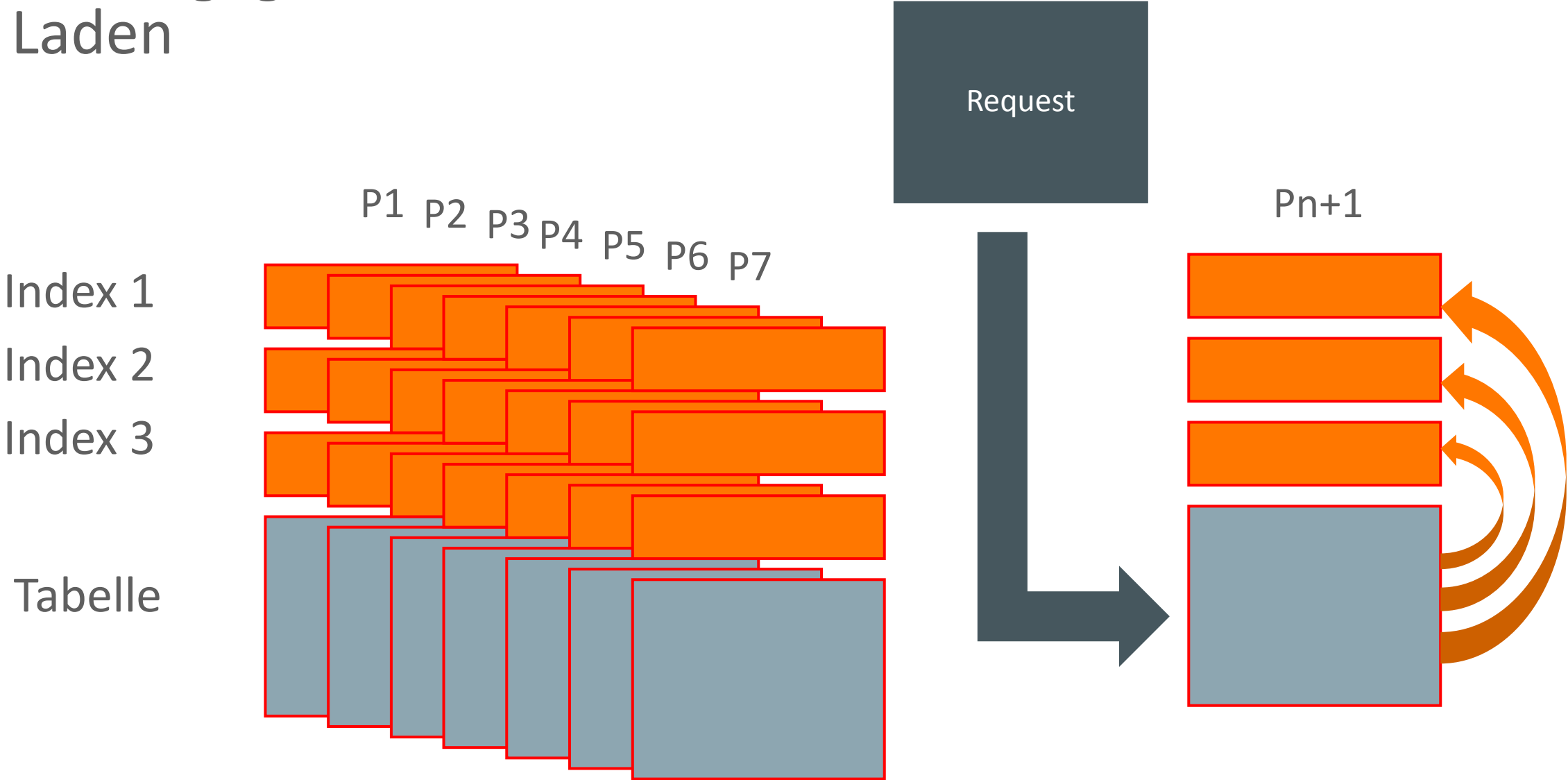
Laden gegen Indices Neue Partition



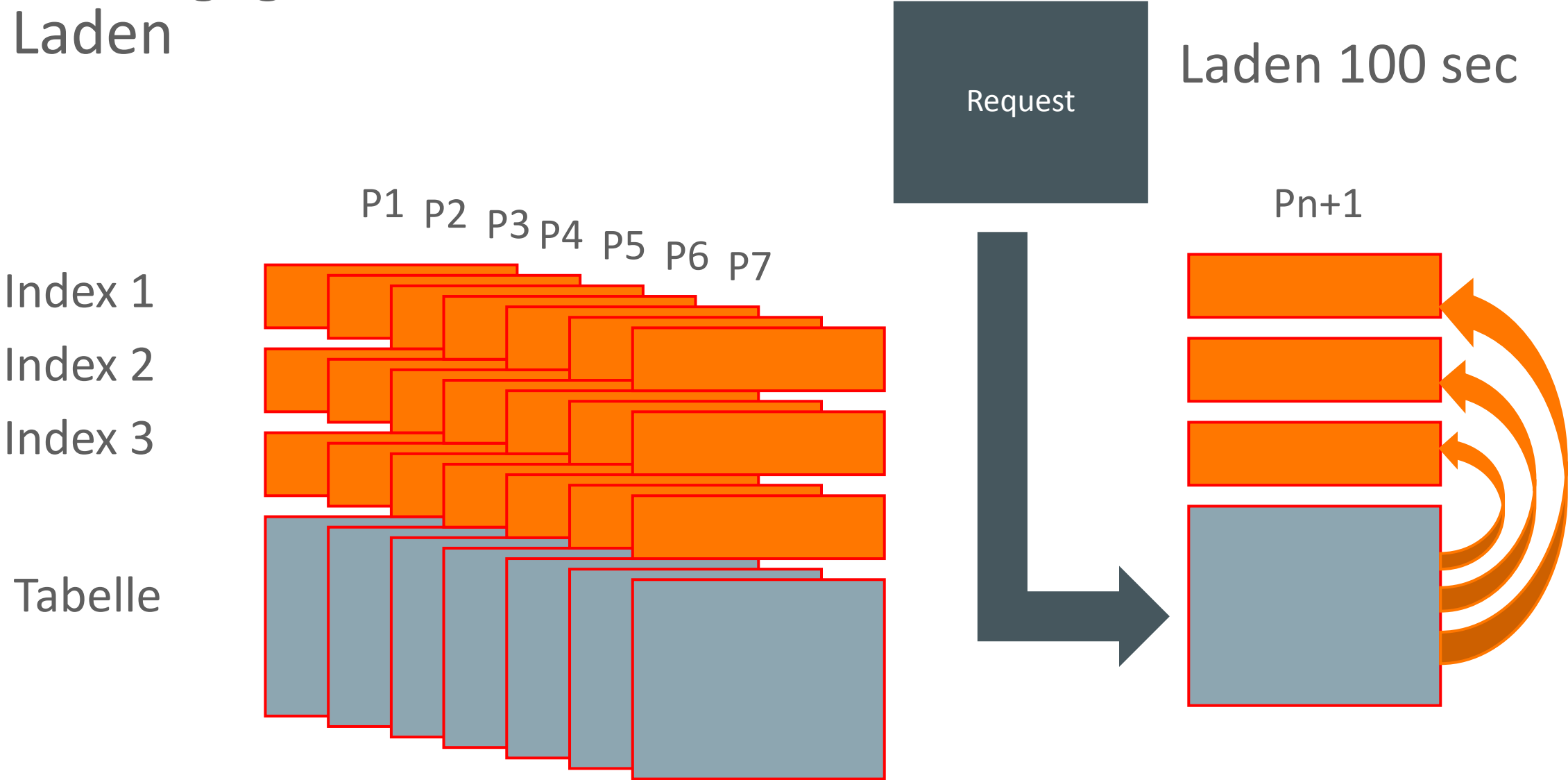
Laden gegen Indices Laden



Laden gegen Indices Laden



Laden gegen Indices Laden



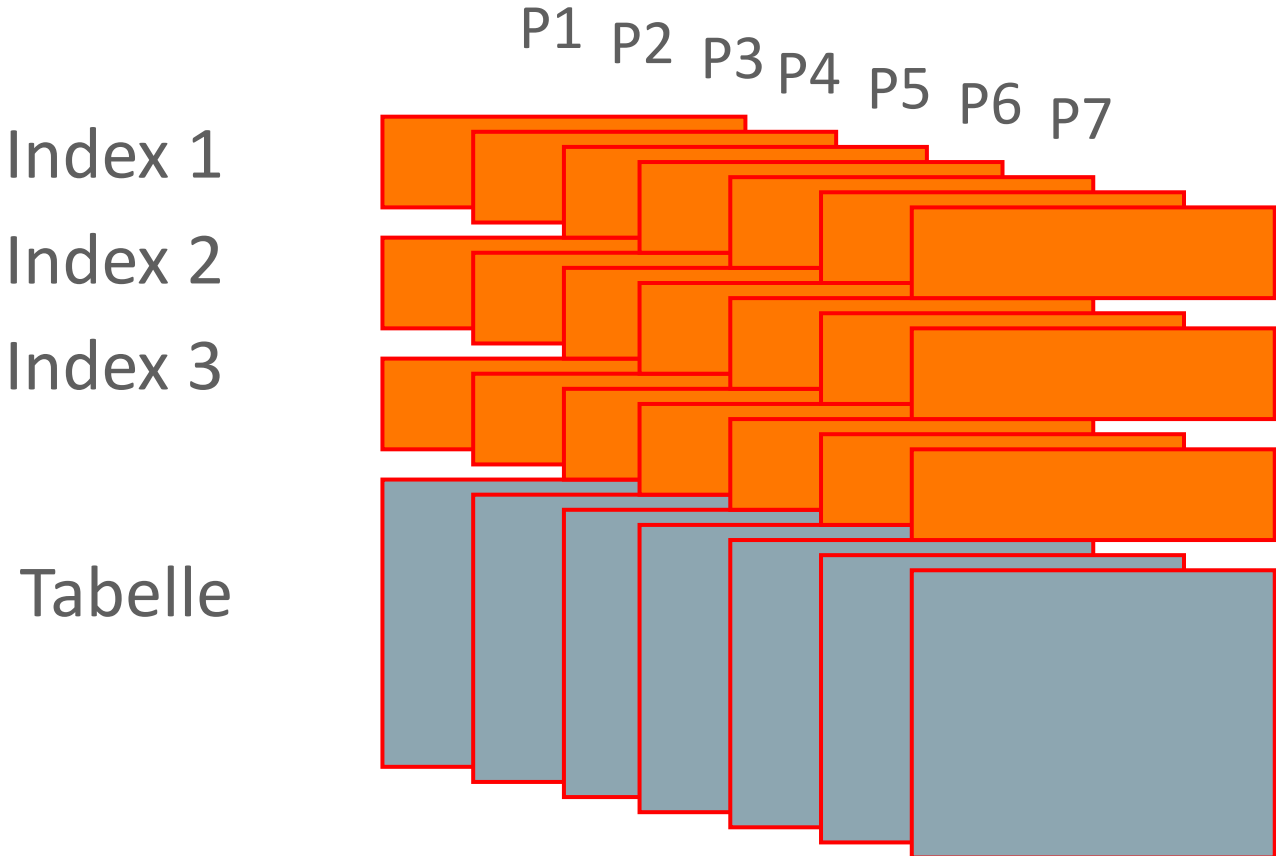
Agenda

- 1 Standard Laden der F-Fakten Tabelle
- 2 Erste Optimierung**
- 3 Unusable Indexes / Table Factoring
- 4 Optimierung mit Hinweis 1842044
- 5 Ausblick

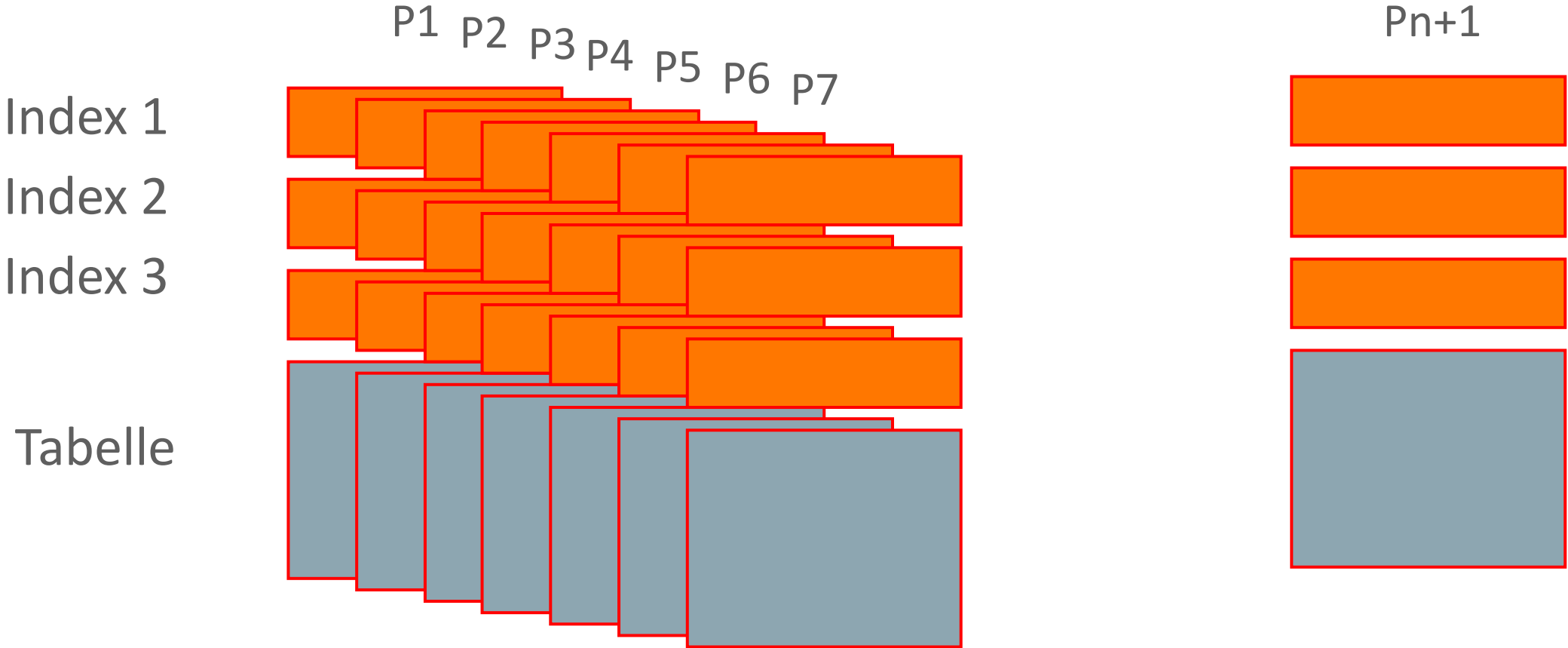
Erste Optimierung

Löschen und Wiederaufbau der Indices

Löschen und Wiederaufbau der Indices

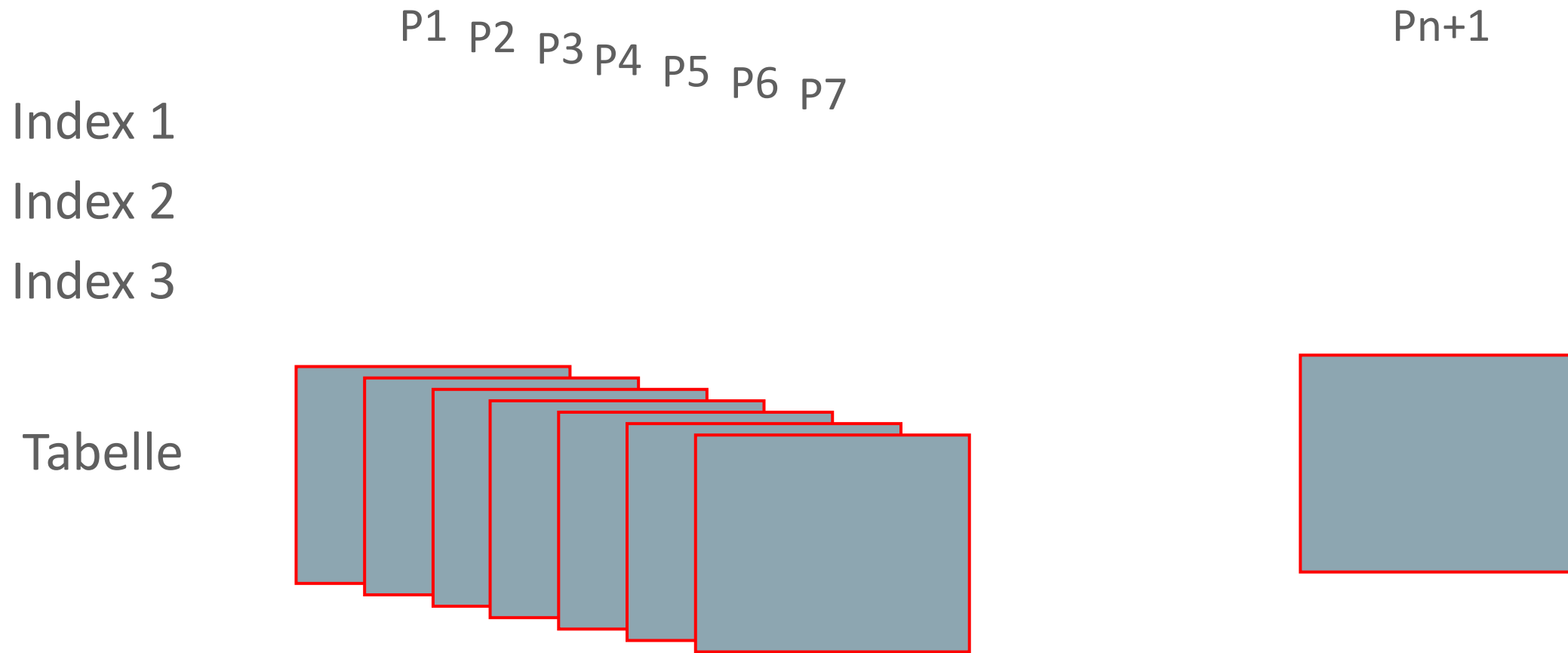


Löschen und Wiederaufbau der Indices Neue Partition



Löschen und Wiederaufbau der Indices

Löschen der Indices



Löschen und Wiederaufbau der Indices

Laden



Laden 6 sec

P1 P2 P3 P4 P5 P6 P7

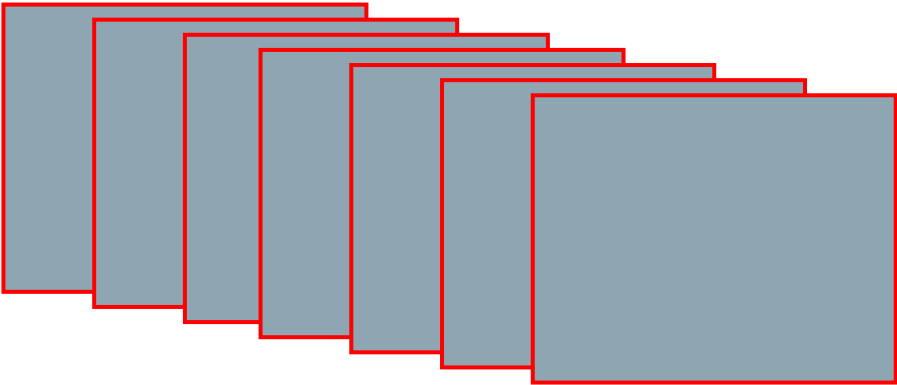
Pn+1

Index 1

Index 2

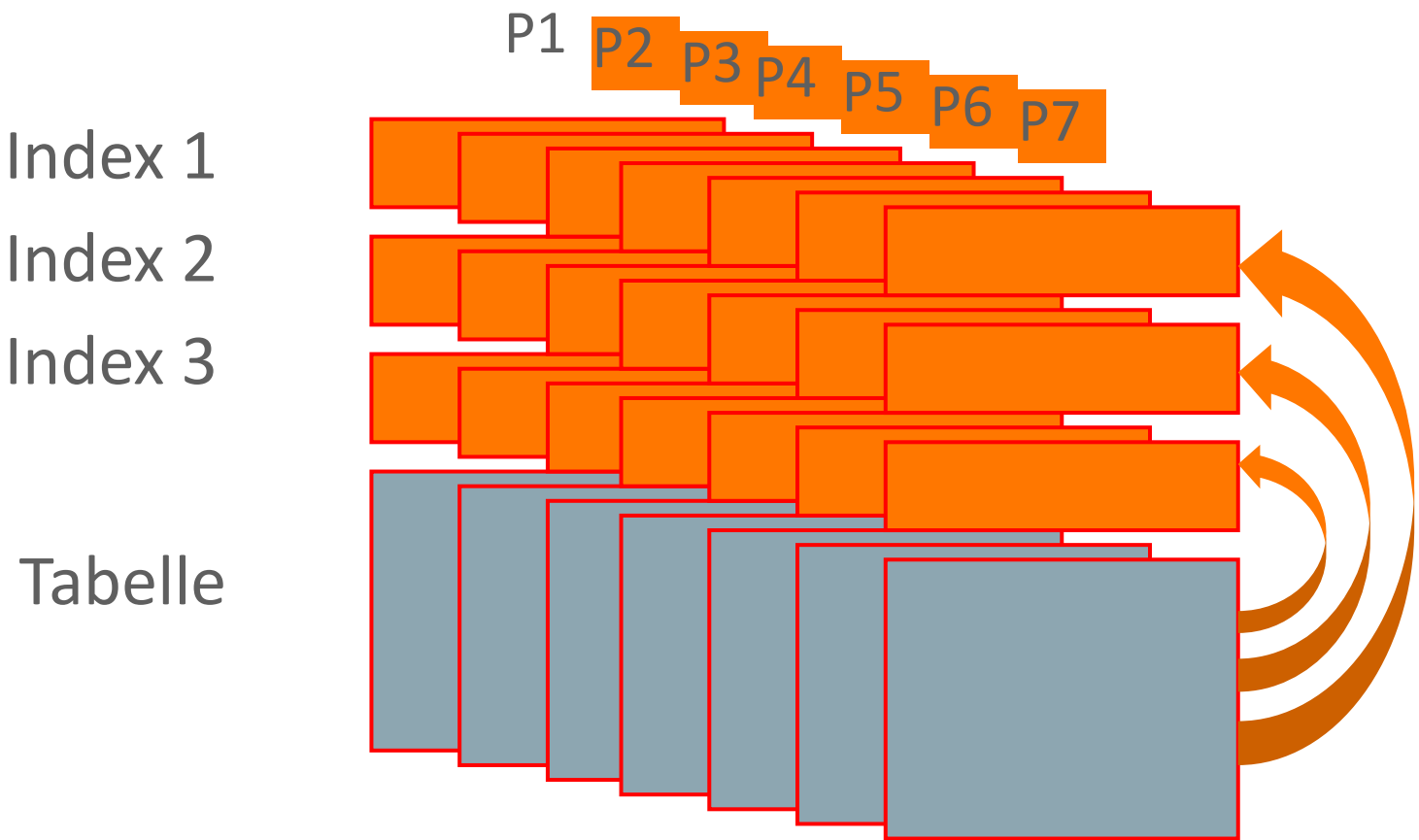
Index 3

Tabelle

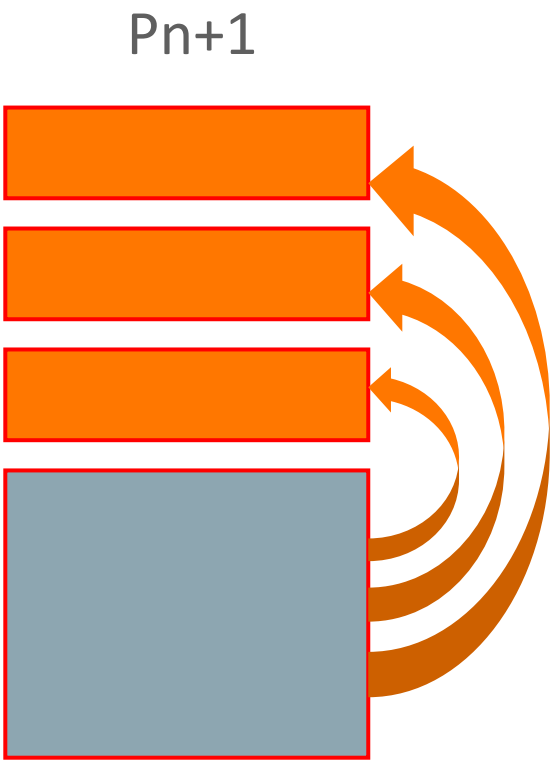


Löschen und Wiederaufbau der Indices

Index Aufbau



Laden 6 sec
Index 200 sec



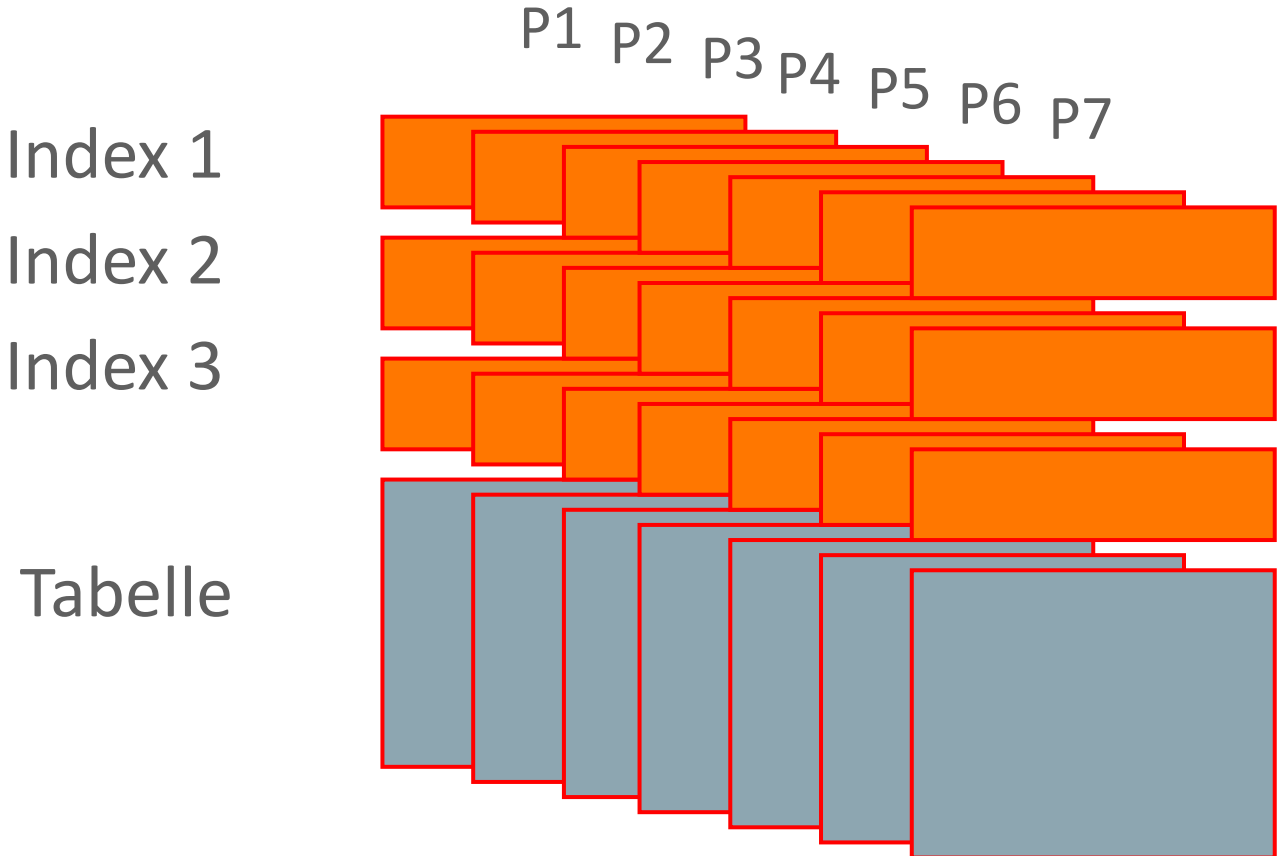
Agenda

- 1 Standard Laden der F-Fakten Tabelle
- 2 Erste Optimierung
- 3 Unusable Indexes / Table Factoring**
- 4 Optimierung mit Hinweis 1842044
- 5 Ausblick

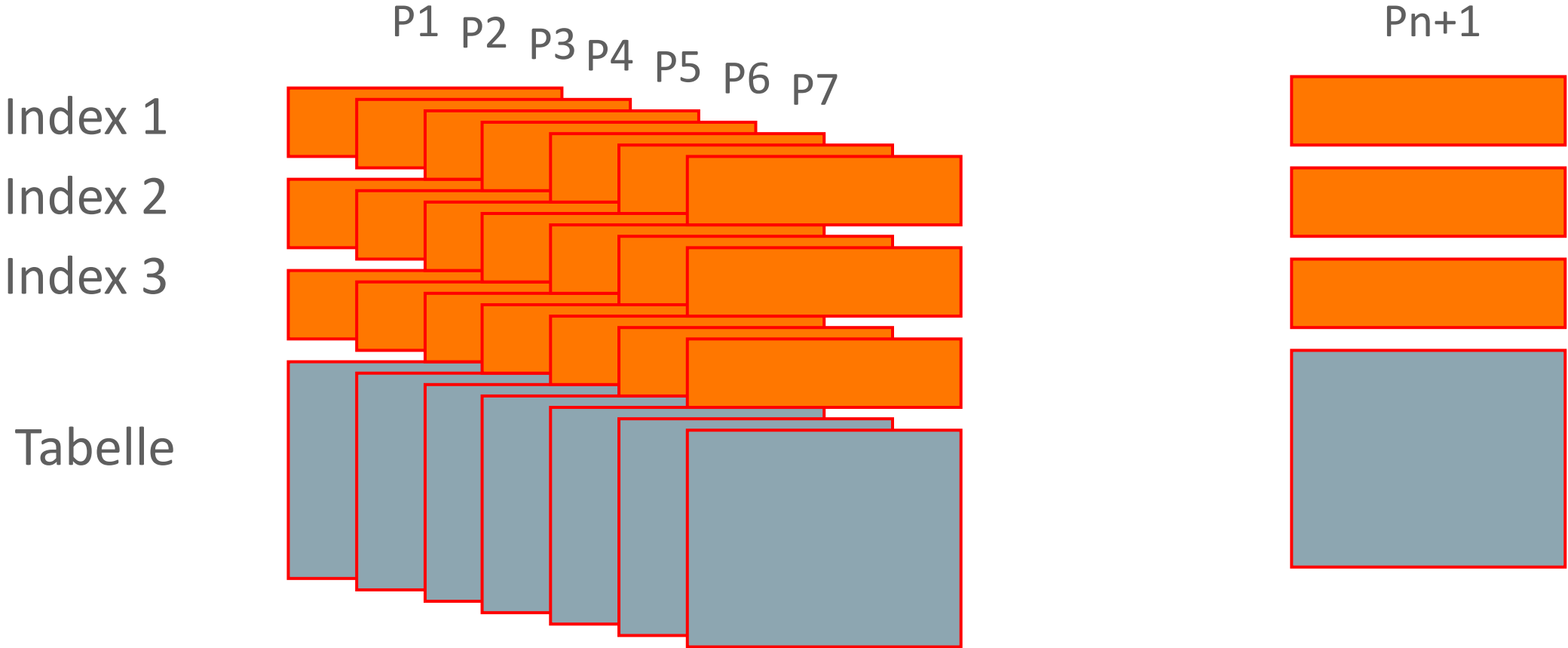
Unusable Indexes / Table Factoring

Unusable Index Partitions

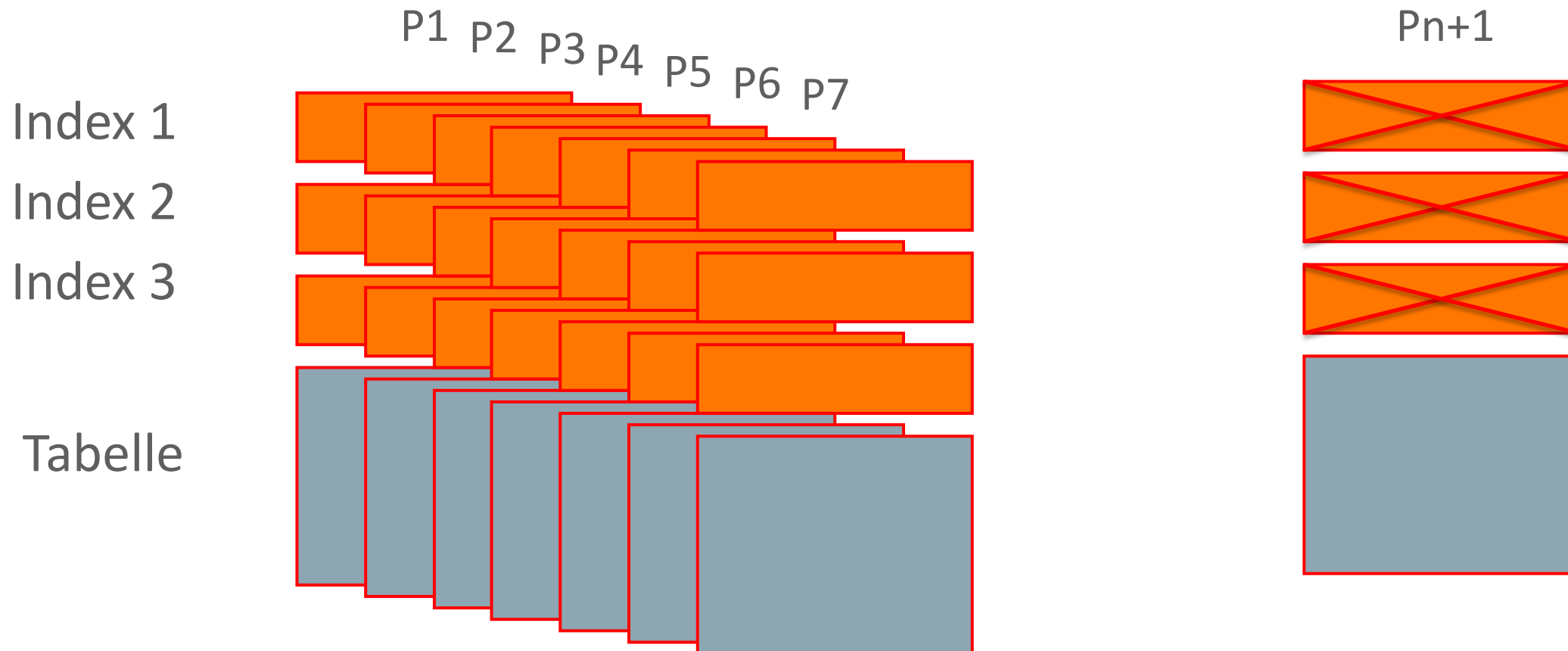
Unusable Index Partition



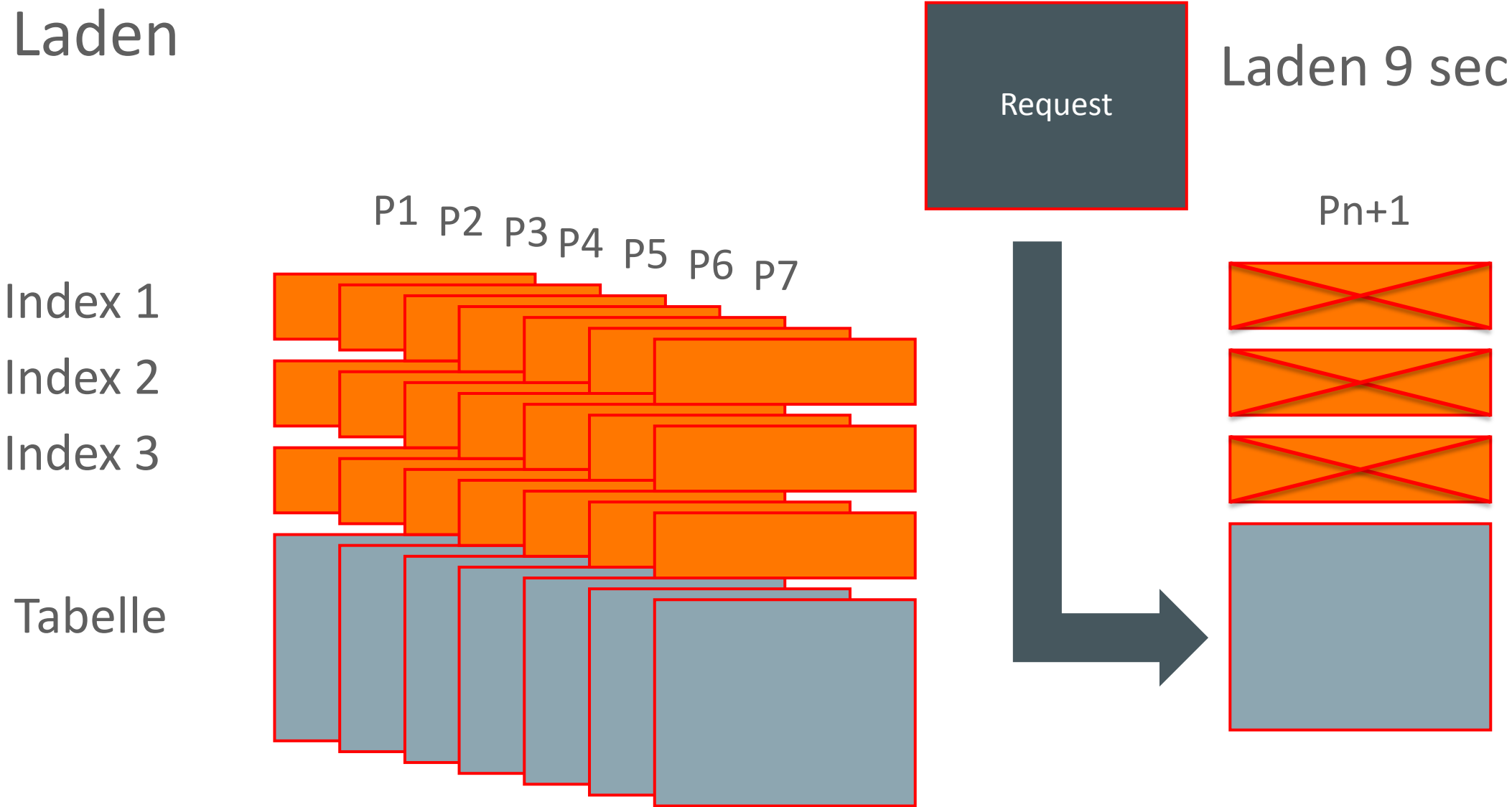
Unusable Index Partition Neue Partition



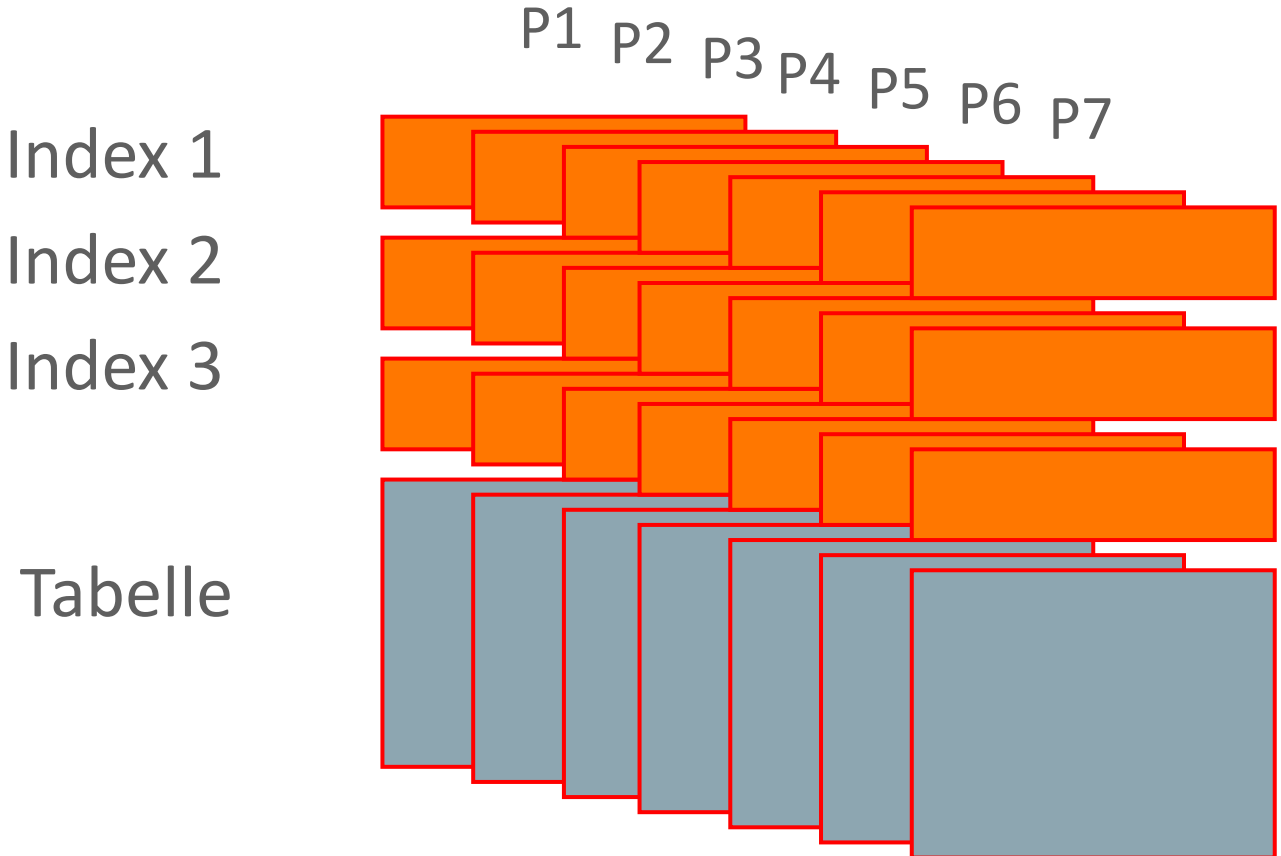
Unusable Index Partition Set unusable



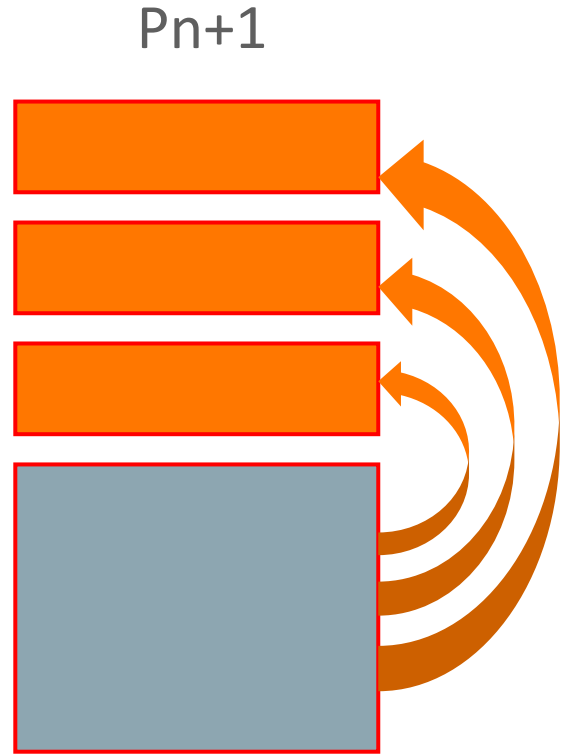
Unusable Index Partition Laden



Unusable Index Partition Rebuild



Laden 9 sec
Index 1,5 sec



Vergleich

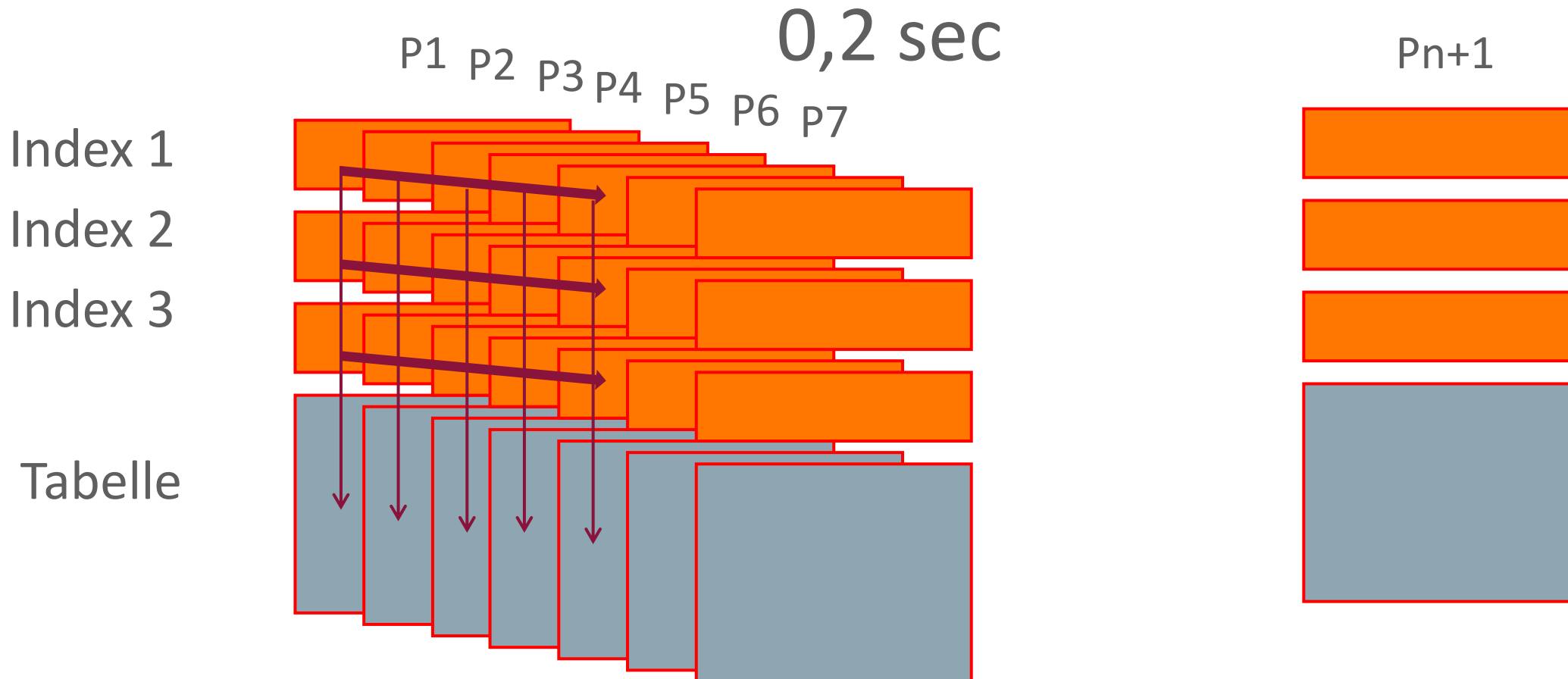
	1 Request	10 Requests
Ohne Optimierung	100 sec	1000 sec
Alle Indices löschen	206 sec	260 sec
Unusable Index Part.	10,5 sec	105 sec

Unusable Indexes / Table Factoring

Query Rewrite mit Table Factoring

Query Rewrite mit Table Factoring Zugriff mit kompletten Indices

```
select count (*) cnt, sum (f.ATR_CTTS) s1  
from "/BIO/FOATR_CR01" F  
where KEY_0ATR_CR01P < 65  
and KEY_0ATR_CR01T between 80 and 100  
and KEY_0ATR_CR012 between 6632 and 7792
```



Query Rewrite mit Table Factoring

Zugriffsplan mit kompletten Indices

Id	Operation	Name
0	SELECT STATEMENT	
1	SORT AGGREGATE	
2	PARTITION RANGE ITERATOR	
3	TABLE ACCESS BY LOCAL INDEX ROWID BATCHED	/BI0/F0ATR_CR01
4	BITMAP CONVERSION TO ROWIDS	
5	BITMAP AND	
6	BITMAP MERGE	
* 7	BITMAP INDEX RANGE SCAN	/BI0/F0ATR_CR01~02
8	BITMAP MERGE	
* 9	BITMAP INDEX RANGE SCAN	/BI0/F0ATR_CR01~01
10	BITMAP MERGE	
* 11	BITMAP INDEX RANGE SCAN	/BI0/F0ATR_CR01~04

Query Rewrite mit Table Factoring Zugriff ohne Indices

```
select count (*) cnt, sum (f.ATR_CTTS) s1  
from "/BIO/FOATR_CR01" F  
where KEY_OATR_CR01P < 65  
and KEY_OATR_CR01T between 80 and 100  
and KEY_OATR_CR012 between 6632 and 7792
```

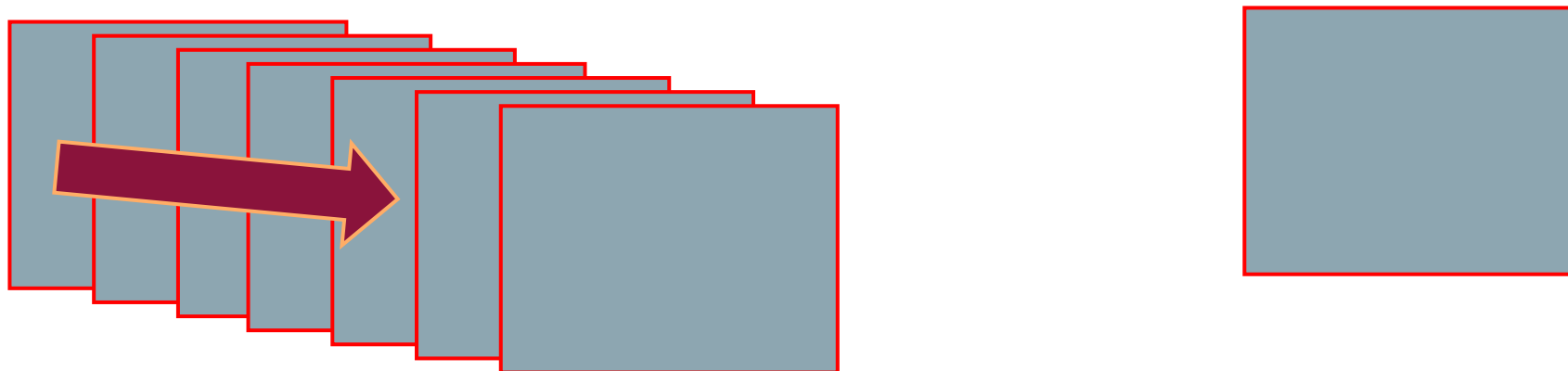
P1 P2 P3 P4 P5 P6 P7 **15 sec**

Index 1

Index 2

Index 3

Tabelle



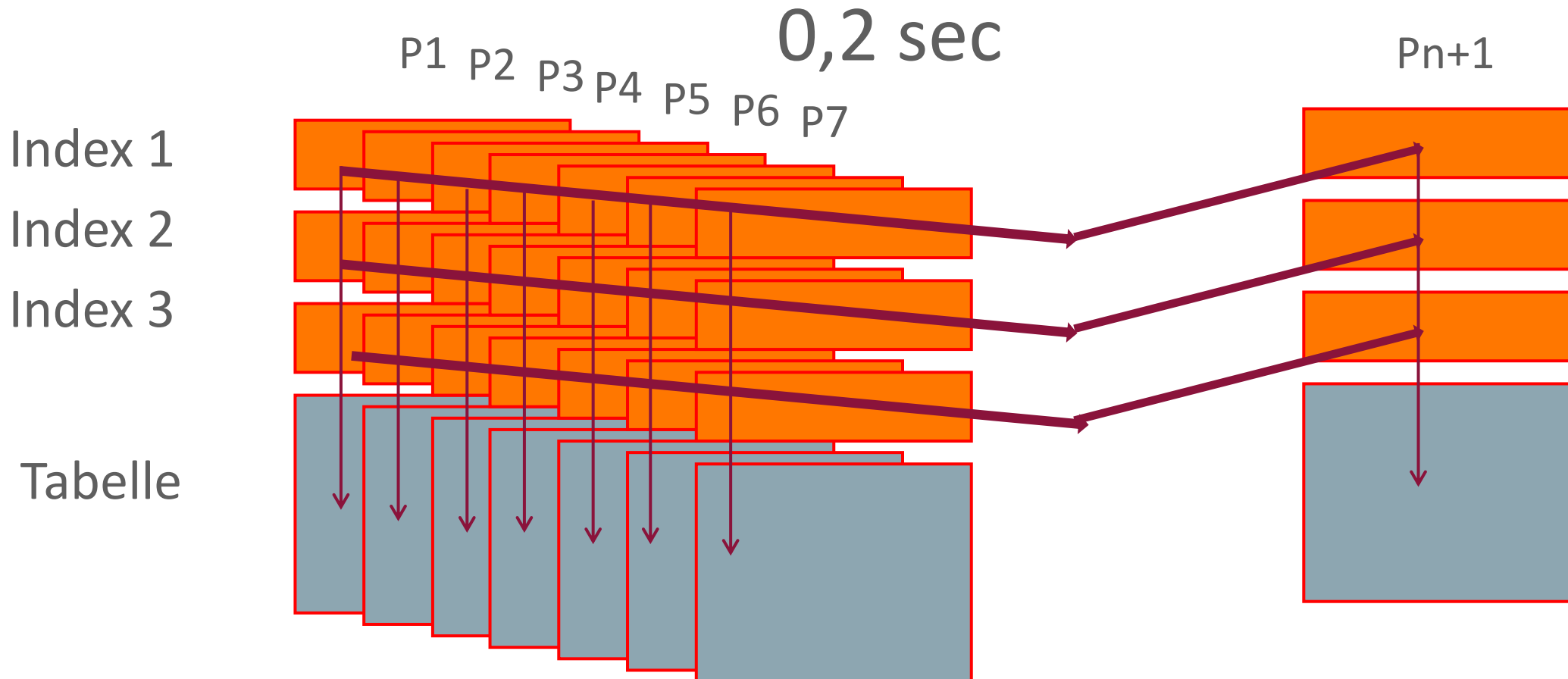
Query Rewrite mit Table Factoring

Zugriffsplan ohne Indices

Id	Operation	Name
0	SELECT STATEMENT	
1	SORT AGGREGATE	
2	PARTITION RANGE ITERATOR	
3	TABLE ACCESS FULL	/BI0/F0ATR_CR01

Query Rewrite mit Table Factoring Zugriff mit kompletten Indices Transactional cube

```
select count (*) cnt, sum (f.ATR_CTTS) s1  
from "/BIO/FOATR_CR01" F  
Where KEY_0ATR_CR01T between 80 and 100  
and KEY_0ATR_CR012 between 6632 and 7792
```



Query Rewrite mit Table Factoring

Zugriff mit kompletten Indices

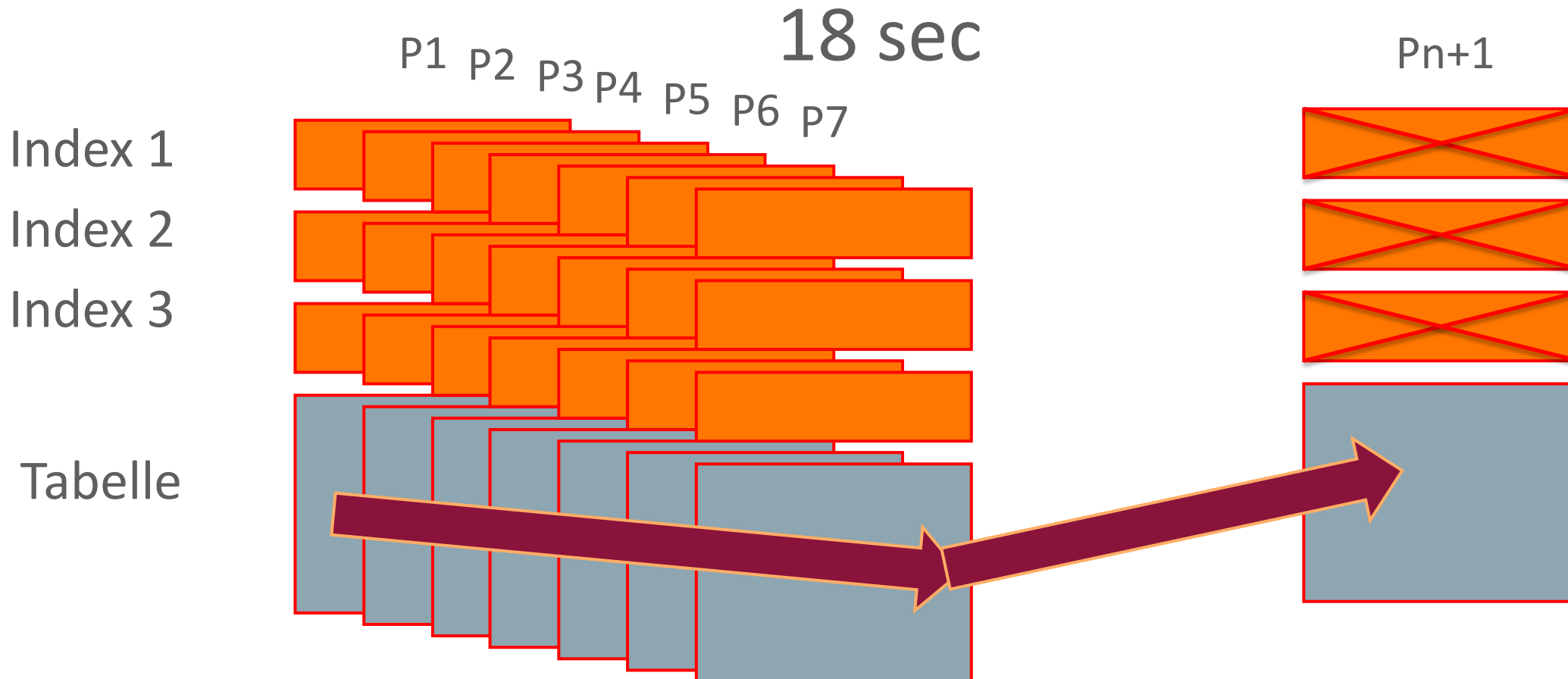
Transactional cube

Id	Operation	Name
0	SELECT STATEMENT	
1	SORT AGGREGATE	
2	PARTITION RANGE ITERATOR	
3	TABLE ACCESS BY LOCAL INDEX ROWID BATCHED	/BI0/F0ATR_CR01
4	BITMAP CONVERSION TO ROWIDS	
5	BITMAP AND	
6	BITMAP MERGE	
* 7	BITMAP INDEX RANGE SCAN	/BI0/F0ATR_CR01~02
8	BITMAP MERGE	
* 9	BITMAP INDEX RANGE SCAN	/BI0/F0ATR_CR01~01
10	BITMAP MERGE	
* 11	BITMAP INDEX RANGE SCAN	/BI0/F0ATR_CR01~04

Query Rewrite mit Table Factoring

Zugriff mit unusable Index Partition

Transactional cube – 10.2



Query Rewrite mit Table Factoring

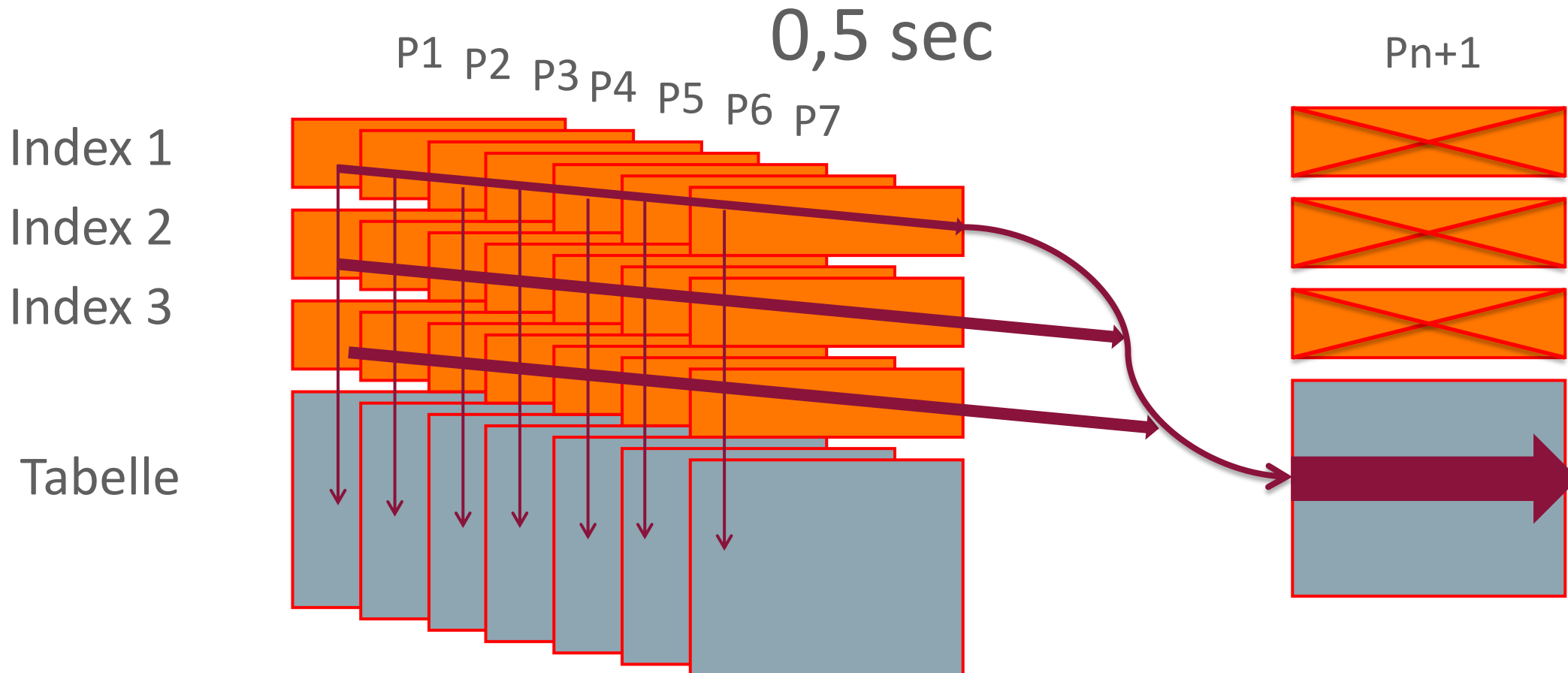
Zugriff mit unusable Index Partition

Transactional cube – 10.2

Id	Operation	Name
0	SELECT STATEMENT	
1	SORT AGGREGATE	
2	PARTITION RANGE ITERATOR	
3	TABLE ACCESS FULL	/BI0/F0ATR_CR01

Query Rewrite mit Table Factoring Zugriff mit unusable Index Partition Transactional cube – 11.2

```
select count (*) cnt, sum (f.ATR_CTTS) s1  
from "/BIO/FOATR_CR01" F  
Where KEY_0ATR_CR01T between 80 and 100  
and KEY_0ATR_CR012 between 6632 and 7792
```



Query Rewrite mit Table Factoring

Zugriff mit unusable Index Partition

Transactional cube – 11.2

```
Select sum (cnt), sum (s1)
From (select count (*) cnt, sum (f.ATR_CTTS) s1
      from "/BI0/F0ATR_CR01" F
      where PARTITION between 1 and 68
      and KEY_0ATR_CR01T between 80 and 100
      and KEY_0ATR_CR012 between 6632 and 7792
      Union all
      select count (*) cnt, sum (f.ATR_CTTS) s1
      from "/BI0/F0ATR_CR01" F
      where PARTITION = 69
      and KEY_0ATR_CR01T between 80 and 100
      and KEY_0ATR_CR012 between 6632 and 7792)
```

Query Rewrite mit Table Factoring

Zugriff mit unusable Index Partition

Transactional cube – 11.2

Id	Operation	Name
0	SELECT STATEMENT	
1	SORT AGGREGATE	
2	VIEW	VW_TE_2
3	UNION-ALL	
4	PARTITION RANGE ITERATOR	
5	TABLE ACCESS BY LOCAL INDEX ROWID BATCHED	/BI0/F0ATR_CR01
6	BITMAP CONVERSION TO ROWIDS	
7	BITMAP AND	
8	BITMAP MERGE	
9	BITMAP INDEX RANGE SCAN	/BI0/F0ATR_CR01~02
10	BITMAP MERGE	
11	BITMAP INDEX RANGE SCAN	/BI0/F0ATR_CR01~01
12	BITMAP MERGE	
13	BITMAP INDEX RANGE SCAN	/BI0/F0ATR_CR01~04
14	PARTITION RANGE SINGLE	
15	TABLE ACCESS FULL	/BI0/F0ATR_CR01

Agenda

- 1 Standard Laden der F-Fakten Tabelle
- 2 Erste Optimierung
- 3 Unusable Indexes / Table Factoring
- 4 Optimierung mit Hinweis 1842044**
- 5 Ausblick

Optimierung mit Hinweis 1842044

Optimierung mit Hinweis 1842044

- Ist in den folgenden Support Packages enthalten:
 - SAP NetWeaver BW 7.30 – SP 10
 - SAP NetWeaver BW 7.31 – SP 8
 - SAP NetWeaver BW 7.40 – SP 3
- Sonst Hinweis 1842044 mit SNOTE einspielen.
- Aktivieren mit:
 - In die Tabelle RSADMIN den folgenden Parameter aufnehmen:
 - OBJECT = ORA_IC_LOAD_NO_DROP VALUE = X

Agenda

- 1 Standard Laden der F-Fakten Tabelle
- 2 Erste Optimierung
- 3 Unusable Indexes / Table Factoring
- 4 Optimierung mit Hinweis 1842044
- 5 Ausblick

Ausblick

**Transactional Cubes mit Bitmap Indices
Compression der geladenen Partitionen**

Ausblick

Transactional Cubes mit Bitmap Indices

- Transactional Cubes können mit Bitmap Indices definiert werden
- Volle Star Query Unterstützung

Ausblick

Compression der geladenen Partitionen

- Laden in unkomprimierte Partition
- Komprimieren bevor die Indizes wieder aufgebaut werden
 - Alter table xx move partition yy compress
- Jeder Block wird nur einmal komprimiert
- Ohne Compression Option möglich.

Pilotkunden????

Fragen???

Joern.Bartels@oracle.com

Hardware and Software Engineered to Work Together

ORACLE®