

# Index- und Zugriffsstrukturen für Data Warehousing

---

JOHANNES ELSMANN, 14INM

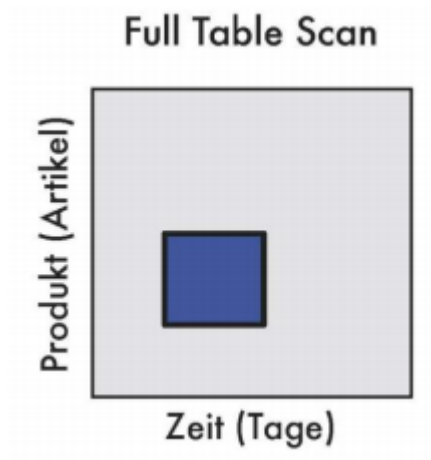
# Inhalt

---

- Wiederholung Vorlesung
- Anlegen von Indexen in Oracle und SQL Server
- Query-Beispiel für Bitmap Index
- Query-Beispiel für Verbundindex (Bitmap Join Index)
- Vergleich mit SQL Server
- Entscheidungskriterien für Index-Auswahl

# Wiederholung

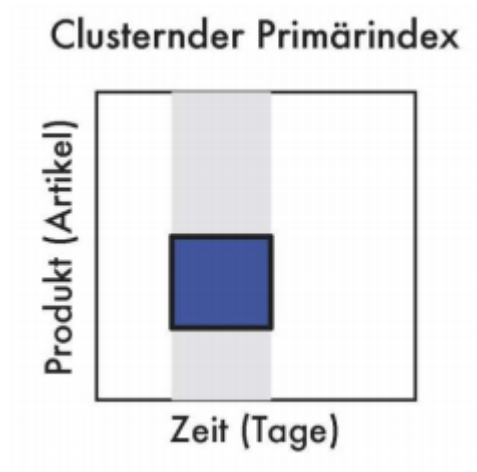
---



- Liest alle Werte
- Sehr teuer

# Wiederholung

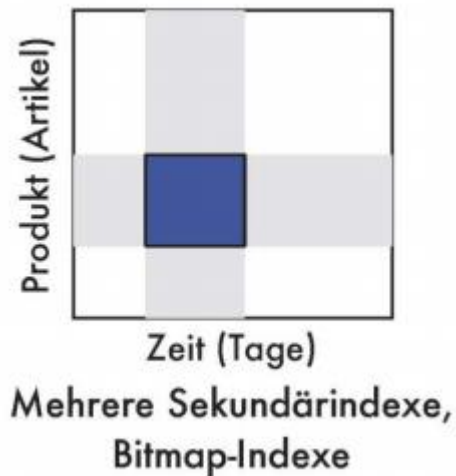
---



- Beschränkt sich auf Bereich in Dimension
- Nur angefragte Tage aber für alle Produkte

# Wiederholung

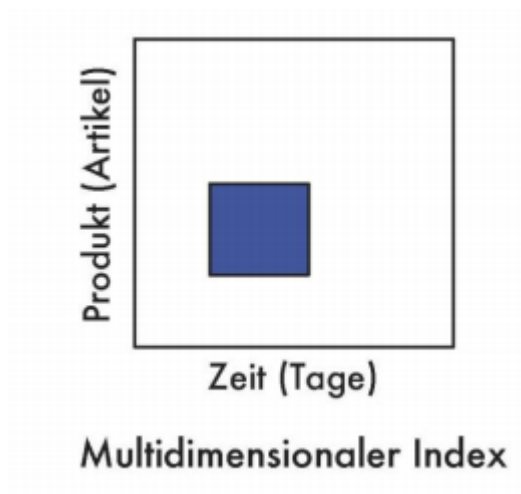
---



- Auswertung eines Bereichs in jeder Dimension
- Bildung der Schnittmenge
- Nicht direkt auf Tupeln sondern ihren TIDs

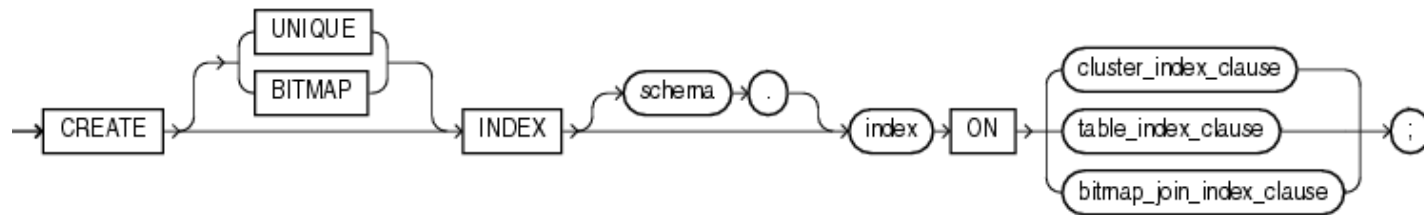
# Wiederholung

---

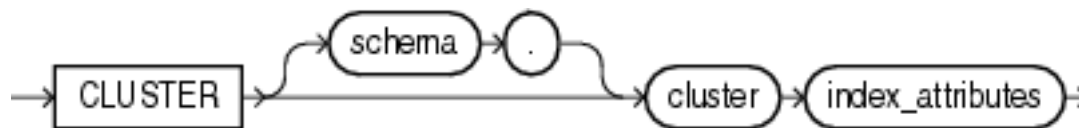


- Liefert direkt den gesuchten Bereich, keine weiteren Auswertungen notwendig

# Index-Deklaration Oracle



- Verweis auf Cluster in Index-Deklaration:



# Index-Deklaration T-SQL

---

```
CREATE [ UNIQUE ] [ CLUSTERED | NONCLUSTERED ] INDEX index_name
    ON <object> ( column [ ASC | DESC ] [ ,...n ] )
    [ INCLUDE ( column_name [ ,...n ] ) ]
    [ WHERE <filter_predicate> ]
    [ WITH ( <relational_index_option> [ ,...n ] ) ]
    [ ON { partition_scheme_name ( column_name )
        | filegroup_name
        | default
        }
    ]
    [ FILESTREAM_ON { filestream_filegroup_name | partition_scheme_name | "NULL" } ]

[ ; ]
```

# Beispiel Bitmap Index (Single Table)

```
SELECT cust_id, cust_gender, cust_marital_status, cust_income_level  
FROM customers;
```

| ID  | Gender | Material Status | Income Level         |
|-----|--------|-----------------|----------------------|
| ... |        |                 |                      |
| 70  | F      | single          | E: 70,000 - 89,999   |
| 80  | F      | married         | C: 150,000 - 169,999 |
| 90  | M      | married         | W: 130,000 - 149,999 |
| 100 | F      | single          | W: 130,000 - 149,999 |
| 110 | F      | single          | C: 150,000 - 169,999 |
| 120 | M      | married         | C: 150,000 - 169,999 |
| ... |        |                 |                      |

# Beispiel Bitmap Index (Single Table)

```
SELECT COUNT(*) FROM customers
WHERE cust_marital_status = 'married'
AND cust_income_level IN
('C: 150,000 - 169,999', 'W: 130,000 - 149,999');
```

| status =<br>'married' | region =<br>'central' | region =<br>'west' |
|-----------------------|-----------------------|--------------------|
| 0                     | 0                     | 0                  |
| 1                     | 1                     | 0                  |
| 1                     | 0                     | 1                  |
| 0                     | 0                     | 1                  |
| 0                     | 1                     | 0                  |
| 1                     | 1                     | 0                  |

| status =<br>'married' | region =<br>'central' | region =<br>'west' |
|-----------------------|-----------------------|--------------------|
| 0                     | 0                     | 0                  |
| 1                     | 1                     | 0                  |
| 1                     | 0                     | 1                  |
| 0                     | 0                     | 1                  |
| 0                     | 1                     | 0                  |
| 1                     | 1                     | 0                  |

| status =<br>'married' | region =<br>'central' | region =<br>'west' |
|-----------------------|-----------------------|--------------------|
| 0                     | 0                     | 0                  |
| 1                     | 1                     | 0                  |
| 1                     | 0                     | 1                  |
| 0                     | 0                     | 1                  |
| 0                     | 1                     | 0                  |
| 1                     | 1                     | 0                  |

# Beispiel Bitmap Join Index (two or more tables)

---

```
SELECT time_id, cust_id, amount_sold FROM sales;
```

| Time ID   | Customer ID | Amount Sold |
|-----------|-------------|-------------|
| 01-JAN-98 | 29700       | 2291        |
| 01-JAN-98 | 3380        | 114         |
| 01-JAN-98 | 67830       | 553         |
| 01-JAN-98 | 179339      | 1           |
| 01-JAN-98 | 127520      | 195         |
| 01-JAN-98 | 33030       | 280         |
| ...       |             |             |

# Beispiel Bitmap Join Index (two or more tables)

---

- Bitmap Index auf einen durchgeführten Join
- Erspart Auswertung, da Join Ergebnis bekannt
- Beispiel: Spalte cust\_gender wird mit Tabelle sales gejoined
- Join Bedingung bei Index-Erstellung angegeben

```
CREATE BITMAP INDEX sales_cust_gender_bjix  
ON sales(customers.cust_gender)  
FROM sales, customers  
WHERE sales.cust_id = customers.cust_id  
LOCAL NOLOGGING COMPUTE STATISTICS;
```

# Beispiel Bitmap Join Index (two or more tables)

```
SELECT sales.time_id, customers.cust_gender, sales.amount_sold  
FROM sales, customers  
WHERE sales.cust_id = customers.cust_id;
```

| Time ID   | Gender | Amount Sold |
|-----------|--------|-------------|
| 01-JAN-98 | M      | 2291        |
| 01-JAN-98 | F      | 114         |
| 01-JAN-98 | M      | 553         |
| 01-JAN-98 | M      | 1           |
| 01-JAN-98 | M      | 195         |
| 01-JAN-98 | M      | 280         |
| ...       |        |             |

# Beispiel Bitmap Join Index (two or more tables)

---

- Ergebnis Join Index wie folgt:

|                | <b>cust_gender = „M“</b> | <b>cust_gender = „F“</b> |
|----------------|--------------------------|--------------------------|
| sales record 1 | 1                        | 0                        |
| sales record 2 | 0                        | 1                        |
| sales record 3 | 1                        | 0                        |
| sales record 4 | 1                        | 0                        |
| sales record 5 | 1                        | 0                        |
| sales record 6 | 1                        | 0                        |
| ...            |                          |                          |

# Indexierung in SQL Server

---

- Zu berücksichtigende Charakteristiken:
  - Clustered vs nonclustered
  - Unique vs nonunique
  - Single column versus multicolumn
  - column order in Index
- Bitmap Indexe existieren nicht
- Durch materialisierte Sichten (vermutlich) kein Bedarf an Bitmap-Join-Index
- Motto: Entspann dich, ich mach das schon.
  - Query Optimizer entscheidet intern über ähnliche Formen (Bitmap Filtering In-Memory)
  - Index Intersection und Index Union intern verarbeitet

# Auswahlkriterien für Index- und Zugriffsstrukturen

---

## Clusterbildung:

- Für häufige offensichtliche Gruppen
  - Bsp.: Employee LastName, FirstName, MiddleName
- Bei hoher Kardinalität (unique oder größtenteils distinct)
- Für hauptsächlich sequentiell angesprochene Werte

## Standard-Index (B-Baum):

- Hohe Kardinalität
- Für gezielte Anfragen auf kleinen Bereichen
- Wenn Speicherplatz keine Rolle spielt

# Auswahlkriterien für Index- und Zugriffsstrukturen

---

## Bitmap-Index:

- Wenig Inserts/Updates (Index update kostet)
- Bei niedriger Kardinalität
- Bei ansprechen mehrerer Spalten (Leichte Kombination)

## Verbundindex oder materialisierte Sicht:

- Hauptsächlich lesender Zugriff mit vielen Join Operationen

# Quellen

---

- Prof. Kudraß, Vorlesungsfolien Indexstrukturen für Data Warehouse, S. 4
- <https://technet.microsoft.com/en-us/ms188783>
- [http://docs.oracle.com/cd/B19306\\_01/server.102/b14200/statements\\_5010.htm](http://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_5010.htm)
- [http://docs.oracle.com/cd/B28359\\_01/server.111/b28313/indexes.htm#i1006549](http://docs.oracle.com/cd/B28359_01/server.111/b28313/indexes.htm#i1006549)
- [https://technet.microsoft.com/de-de/library/ms191523\(v=sql.105\).aspx](https://technet.microsoft.com/de-de/library/ms191523(v=sql.105).aspx)
- [https://msdn.microsoft.com/en-us/library/ms190639\(SQL.90\).aspx](https://msdn.microsoft.com/en-us/library/ms190639(SQL.90).aspx)