



## Partitionierung

DOAG Datenbanken  
03.06.2014, Düsseldorf

Klaus Reimers  
[info@ordix.de](mailto:info@ordix.de)  
[www.ordix.de](http://www.ordix.de)

- Intention
- Basisarten
  - Range Partitioning
  - Hash Partitioning
  - List Partitioning
  - System Partitoning
- Subarten
  - Composite Partitioning
  - Virtual Column Based Partitioning
  - Partitioning by Reference
  - Interval Partitioning
- Index Partitioning
- Neuerungen Oracle 12c
- Sonstiges
- Fazit

- Kundenprojekte
  - Performance verbessern
  - Kosten senken
  - billigeren Storage einsetzen
  - große Datenmengen besser handhaben
- Fragestellungen
  - Wann lohnt sich Partitionierung?
  - Wieso die Kosten senken?
    - Partitioning ist doch kostenpflichtig!
  - Performance verbessern
    - Wann und wie geht das?

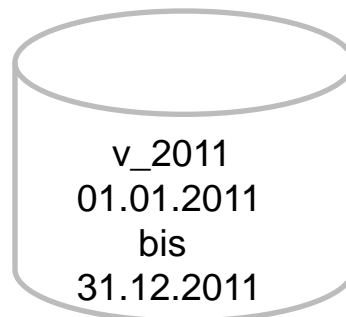
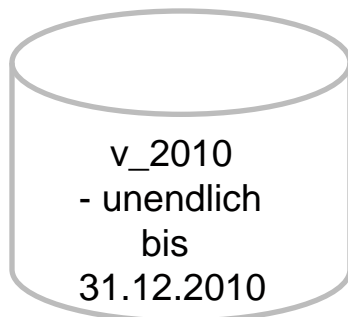


- Intention
- Basisarten
  - Range Partitioning
  - Hash Partitioning
  - List Partitioning
  - System Partitoning
- Subarten
  - Composite Partitioning
  - Virtual Column Based Partitioning
  - Partitioning by Reference
  - Interval Partitioning
- Index Partitioning
- Neuerungen Oracle 12c
- Sonstiges
- Fazit

- Hauptverfahren
  - Range (seit Oracle 8.0)
  - Hash (seit Oracle 8.1)
  - List (seit Oracle 9.0)
  - System (seit Oracle 11.1)
- Composite Partitioning (seit Oracle 8.1)
- erweiterte Verfahren (seit Oracle 11.1)
  - Interval (verfügbar mit Range)
  - REF (verfügbar mit Range, Hash, List)
  - Virtual Column Based (verfügbar mit Range, Hash, List)
- diverse kleine Erweiterungen (Oracle 12c)

```
CREATE TABLE verkauf (  
rechnungsnr      number,  
jahr             number(4) NOT NULL,  
monat           number(2) NOT NULL,  
tag             number(2) NOT NULL)  
PARTITION BY RANGE (jahr)  
  ( PARTITION v_2010 VALUES LESS THAN (2011)  
    TABLESPACE ts1,  
    PARTITION v_2011 VALUES LESS THAN (2012)  
    TABLESPACE ts2  
  );
```

Bei Verkaufsdaten ergibt sich folgende Partitionierung:



- Hash-Funktion für Partitionierungsspalten
- Geeignet für Tabellen, die keine eindeutige Partitionierungsspalte haben
- Verteilung aufgrund Hash-Algorithmus
- Partitionen haben für Benutzer keine logische Bedeutung

```
CREATE TABLE PRODUKT
    ( PROD_NR NUMBER ( 6 ) ,
      PROD_NAME      VARCHAR2 ( 50 )
    )
STORAGE ( INITIAL 20M )
PARTITION BY HASH ( PROD_NR )
PARTITIONS 16
STORE IN ( TBS1 , TBS2 , TBS3 , TBS4 );
```

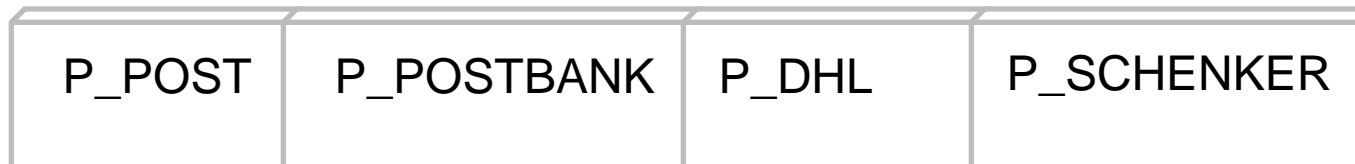
- nicht in Listen definierte Werte:
  - 9.0: nicht möglich
  - 9.2: über DEFAULT eigene Partition
- keine Multicolumn-Partitionierung
- je Liste muss mindestens ein Wert enthalten sein

```
CREATE TABLE ordix_ma
    (pers_nr      NUMBER,
     name         VARCHAR2(40),
     geschlecht   CHAR(1))
PARTITION BY LIST (geschlecht)
    (PARTITION geschlecht_m VALUES ('M'),
     PARTITION geschlecht_w VALUES ('W'),
     PARTITION sonstige VALUES (DEFAULT) -- ab 9.2
    );
```












- Einschränkung auf Partitionen kann nur durch Applikation erfolgen
- Datenbank kann keine Aussage zu den Inhalten der Partition treffen
- Mit „System“ ist hier also die Applikation gemeint

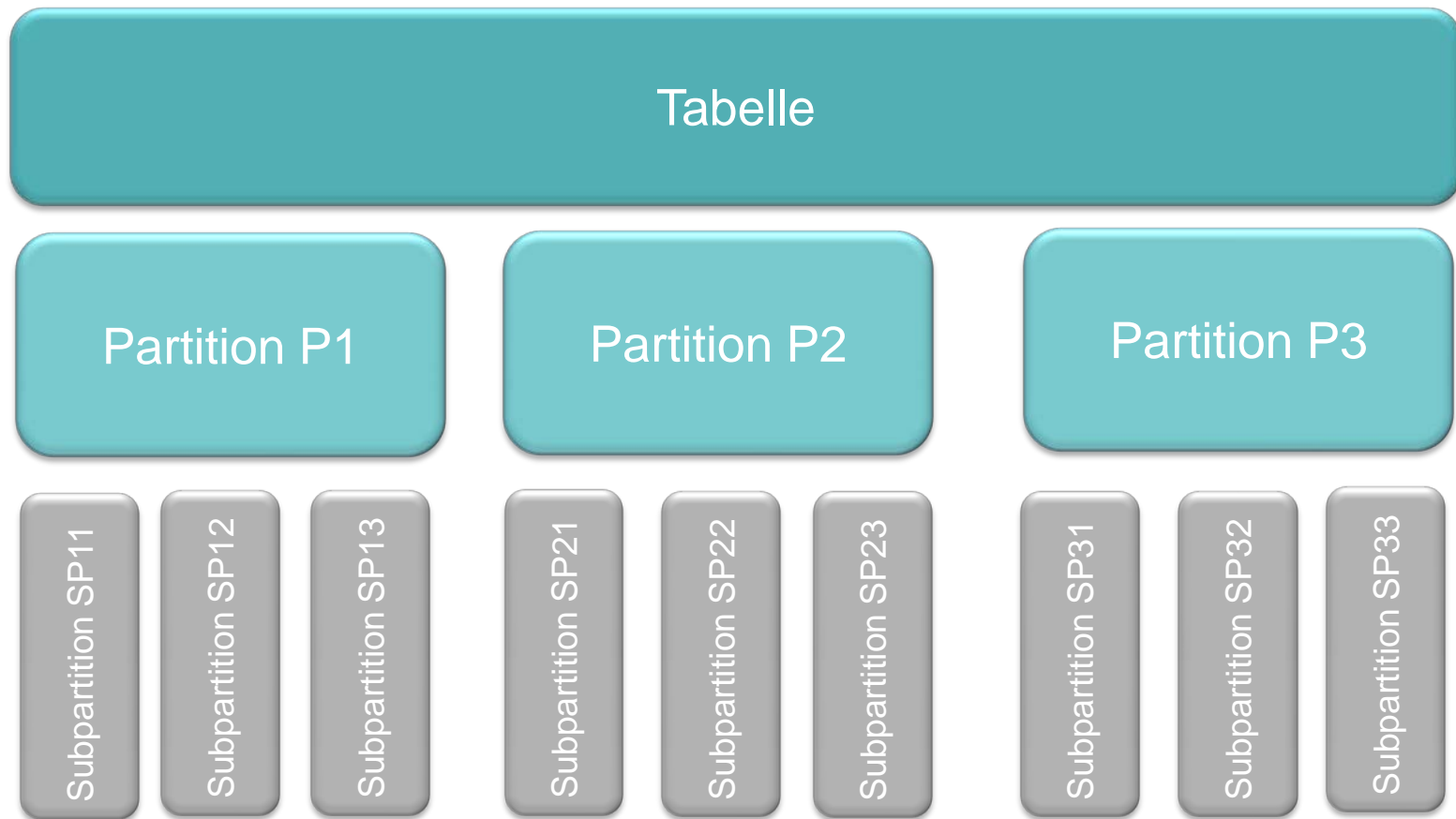
```
CREATE TABLE kunde (name VARCHAR2(60), ...)  
PARTITION BY SYSTEM  
( PARTITION p_post      TABLESPACE tbs_1  
, PARTITION p_postbank TABLESPACE tbs_2  
, PARTITION p_dhl      TABLESPACE tbs_3  
, PARTITION p_schenker TABLESPACE tbs_4  
);
```



- Intention
- Basisarten
  - Range Partitioning
  - Hash Partitioning
  - List Partitioning
  - System Partitoning
- Subarten
  - Composite Partitioning
  - Virtual Column Based Partitioning
  - Partitioning by Reference
  - Interval Partitioning
- Index Partitioning
- Neuerungen Oracle 12c
- Sonstiges
- Fazit

Spaltenliste

erste \ zweite	Range	List	Hash
Range (Intervall)			
List			
Hash			



- Ab 11g virtuelle Spalten auf Tabellen möglich
- Auf Basis virtueller Spalten Partitionierung möglich

Virtuelle  
Spalte

```
CREATE TABLE bevoelkerung
( land          VARCHAR2(60)
, anzahl_frauen NUMBER
, anzahl_maenner NUMBER
, bevoelkerung AS ( anzahl_frauen + anzahl_maenner )
)
PARTITION BY RANGE( bevoelkerung )
( PARTITION p_klein  VALUES LESS THAN (1000000)
, PARTITION p_mittel VALUES LESS THAN (20000000)
, PARTITION p_gross  VALUES LESS THAN (MAXVALUE)
);
```



Partitionierungskriterium  
**DATUM** nur in  
Mastertabelle notwendig!

```
CREATE TABLE warenkorb
->( id          NUMBER
  , datum      DATE
  , summe_preis NUMBER
  , CONSTRAINT warenkorb_pk PRIMARY KEY( id )
 )
PARTITION BY range( DATUM )
( PARTITION p2007 VALUES LESS THAN(to_date( '01.01.2008','DD.MM.YYYY' ))
  , PARTITION p2008 VALUES LESS THAN(to_date( '01.01.2009','DD.MM.YYYY' ))
 );
```

Referenz

```
CREATE TABLE warenkorb_position
( id          NUMBER NOT NULL ENABLE -- keine virt. Spalten
  , pos       NUMBER
  , art_nr    NUMBER
-> , CONSTRAINT warenkorb_pos_fk FOREIGN KEY( id )
                                REFERENCES warenkorb( id )
 ) PARTITION BY REFERENCE( warenkorb_pos_fk );
```

- Datentyp der Partition Key Column: NUMBER oder DATE
- Interval-Funktion bestimmt die Grenzen neuer Partitionen
- Interval-Funktion bestimmt den Datenbereich
- Es können und dürfen „Lücken“ entstehen

```
CREATE TABLE daten( datum date, x number, ... )  
PARTITION BY RANGE (datum)  
INTERVAL( NUMTOYMINTERVAL( 1, 'year'))  
( PARTITION P2006 values less than (TO_DATE('01.01.2007', 'DD.MM.YYYY'))  
, PARTITION P2007 values less than (TO_DATE('01.01.2008', 'DD.MM.YYYY'))  
, PARTITION P2008 values less than (TO_DATE('01.01.2009', 'DD.MM.YYYY'))  
, PARTITION P2009 values less than (TO_DATE('01.01.2010', 'DD.MM.YYYY'))  
);
```

```
INSERT INTO xxx VALUES( '11.09.2011',... )
```



- Intention
- Basisarten
  - Range Partitioning
  - Hash Partitioning
  - List Partitioning
  - System Partitoning
- Subarten
  - Composite Partitioning
  - Virtual Column Based Partitioning
  - Partitioning by Reference
  - Interval Partitioning
- Index Partitioning
- Neuerungen Oracle 12c
- Sonstiges
- Fazit

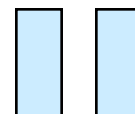
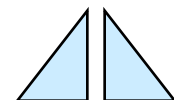
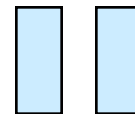
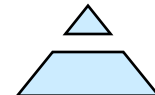
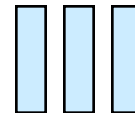
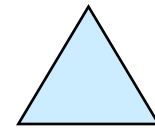
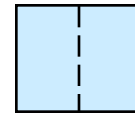


Oracle unterstützt folgende Indextypen

- Nonpartitioned Indexes
- Global prefixed Indexes
- Local prefixed Indexes
- Local nonprefixed Indexes

Tabelle

Index



- Intention
- Basisarten
  - Range Partitioning
  - Hash Partitioning
  - List Partitioning
  - System Partitoning
- Subarten
  - Composite Partitioning
  - Virtual Column Based Partitioning
  - Partitioning by Reference
  - Interval Partitioning
- Index Partitioning
- Neuerungen Oracle 12c
- Sonstiges
- Fazit

- Global Index Maintenance
  - Asynchrone Pflege bei DROP und TRUNCATE
- Reference Partitioning
  - Kaskadierende Funktionalität bei EXCHANGE und TRUNCATE
  - Nutzung des INTERVAL Features
- Move Partition
  - Onlinefähigkeit
- Partial Indexes
  - Keine zwingende 1:1 Beziehung mehr
- Maintenance Erweiterungen
  - Verbesserungen bei 3-Step Kommandos

- Intention
- Basisarten
  - Range Partitioning
  - Hash Partitioning
  - List Partitioning
  - System Partitoning
- Subarten
  - Composite Partitioning
  - Virtual Column Based Partitioning
  - Partitioning By Reference
  - Interval Partitioning
- Index Partitioning
- Neuerungen Oracle 12c
- Sonstiges
- Fazit

- Administrative Tätigkeiten
  - Exchange Partition
- Partitioning von LOBS
  - im selben Tablespace
  - in eigenem Tablespace
  - in unterschiedlichen Tablespaces
- Partitioning von IOTs
  - Normale IOT
  - IOT mit Overflow im selben Tablespace
  - Overflow in anderen Tablespaces

- Intention
- Basisarten
  - Range Partitioning
  - Hash Partitioning
  - List Partitioning
  - System Partitoning
- Subarten
  - Composite Partitioning
  - Virtual Column Based Partitioning
  - Partitioning by Reference
  - Interval Partitioning
- Index Partitioning
- Sonstiges
- Fazit

- administrativer Mehraufwand
  - add partition vs. split partition vs. Interval Partitioning
  - unusable Indizes
- Performance
  - Partition Pruning
  - Full Partition Scan
  - Kleinere Indexbäume
  - Vermeidung von Hot Blocks
- Kosten
  - Höhere Lizenzkosten
  - Geringere Kosten durch
    - Verlagerung auf billigeren Storage
    - Minimierung des Backup Volumens



Zentrale Paderborn  
Westernmuer 12 - 16  
33098 Paderborn  
Tel.: 05251 1063-0

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

Zentrales Fax:  
0180 1 67349 0  
0180 1 ORDIX 0

Weitere Geschäftsstellen  
in Köln, Münster und Neu-Ulm

E-Mail: [info@ordix.de](mailto:info@ordix.de)  
Internet: <http://www.ordix.de>

**Vielen Dank für Ihre Aufmerksamkeit!**