

Секционирование данных в СУБД ORACLE

Секционирование

Метод, позволяющий хранить сегмент данных, такой как таблица, в виде нескольких сегментов, сохраняя логическую монолитную структуру



Информация о секционировании

USER, ALL и DBA представления

- ..._TABLES – информация о таблицах
- ..._PART_TABLES - информация о секционированных таблицах
- ..._TAB_PARTITIONS - информация о табличных секциях
- ..._PART_KEY_COLUMNS – информация о ключах секционирования
- ..._SEGMENTS - информация о сегментах хранения
- ..._OBJECTS - информация о объектах БД

Секция – отдельный сегмент

- Разные сегменты могут находиться в разных ТП , а значит на разных дисках
- Различные секции-сегменты при общности логической структуры могут иметь собственные физические атрибуты
- Отдельные ТП находиться в состоянии OFFLINE, не нарушая работоспособности всей таблицы

Физические атрибуты секций

```
CREATE TABLE DEACore.sales
(
  prod_id NUMBER,
  time_id DATE
)
PARTITION BY RANGE (time_id)
(
  PARTITION sales_11q1 VALUES LESS THAN ('01-APR-2011'),
  PARTITION sales_11q2 VALUES LESS THAN ('01-JUL-2011'),
  PARTITION sales_11q3 VALUES LESS THAN ('01-OCT-2011'),
  PARTITION sales_11q4 VALUES LESS THAN ('01-JAN-2012')
);
```

Логические и физические атрибуты

- Каждая секция таблицы имеет общие логические атрибуты
- Имена и количество столбцов, их порядок и типы, ограничения целостности, атрибут ROW MOVEMENT
- Но каждая из секций имеет собственные физические атрибуты ТП, блочные параметры PCTFREE и PCTUSED, атрибуты COMPRESS и NOLOGGING

Ссылка на секцию

FROM PARTITION

- Данные могут быть выбраны непосредственно из секции

```
SELECT * FROM DEACore.sales PARTITION (sales_11q1);
```

- Может быть задана ассоциативная ссылка

```
SELECT * FROM DEACore.sales PARTITION FOR ('02-FEB-2011');
```

Изменение ключа секционирования в данных

- Изменение ключа секционирования может потребовать переноса строки в другую секцию

```
UPDATE DEACore.sales PARTITION(sales_11q1)  
SET time_id = time_id+90;
```

ORA*14402: updating partition key column would cause a partition change

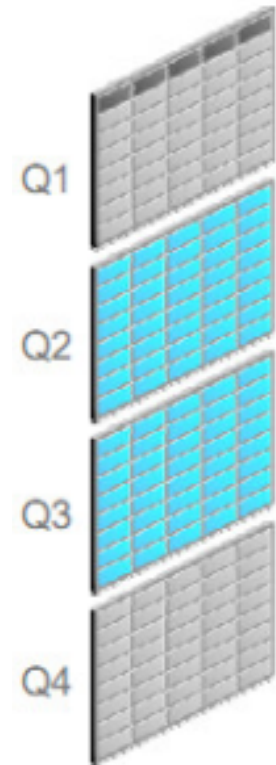
- Выполнение такого переноса нужно явным образом разрешить

```
ALTER TABLE DEACore.sales ENABLE ROW MOVEMENT;  
UPDATE DEACore.sales PARTITION(sales_11q1)  
SET time_id = time_id+90;
```


Исключение секций из выборки

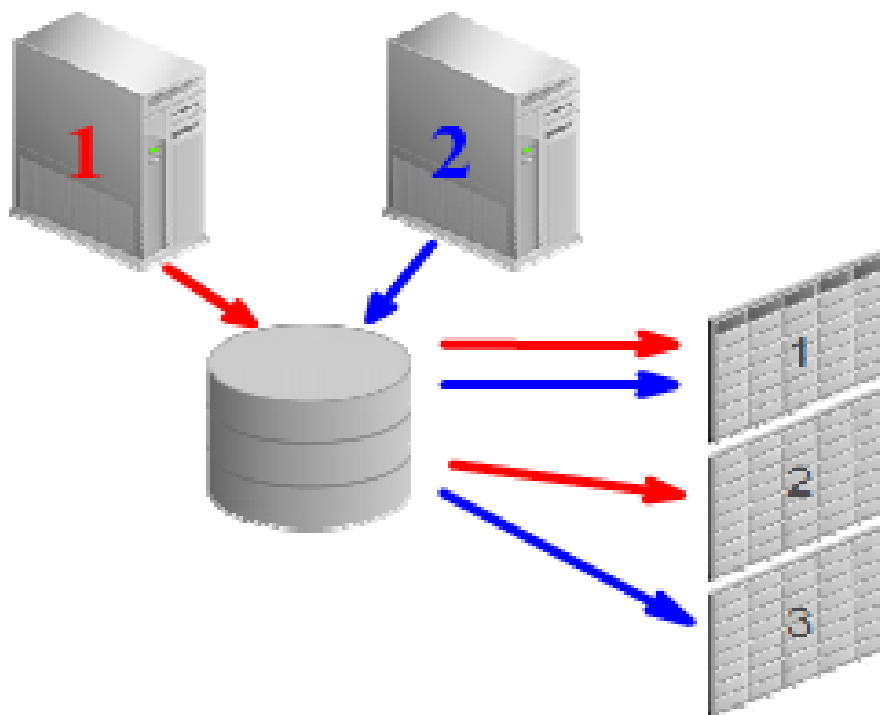
Partition pruning

- Оптимизатор анализирует фразы FROM и WHERE и исключает ненужные секции из просмотра
- Ненужные секции выбираются:
 - Статически** (если значение указано явно)
 - Динамически** (если значение заранее не известно)



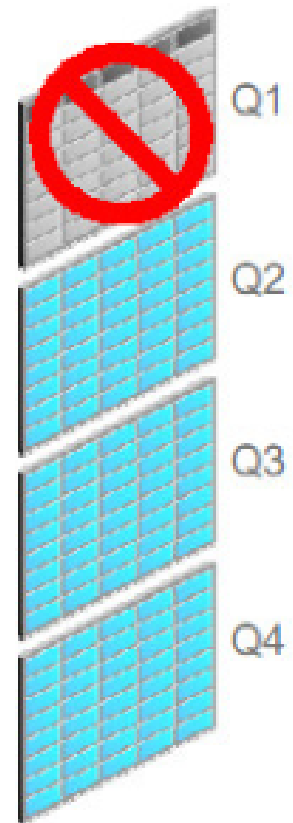
Параллельные операции и секционирование

- Секция – единица параллелизма
- Секции могут сканироваться, загружаться и обновляться параллельно



Секция в ТП OFFLINE

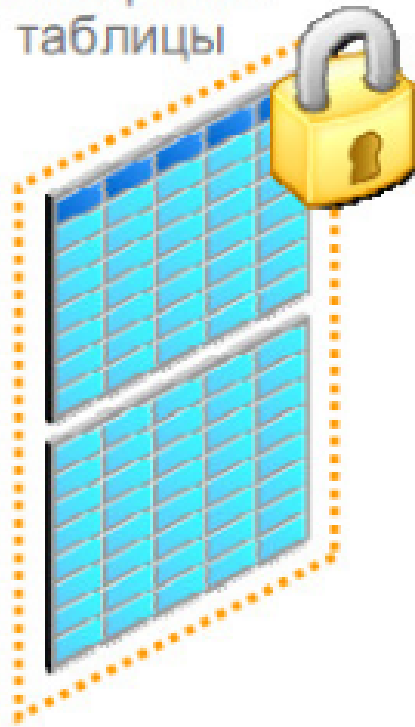
- Из-за сбоя или при обслуживании некоторые ТП могут находиться в состоянии **OFFLINE**
- При попытке чтения табличных данных из такого ТП выдаётся сообщение об ошибке
`SELECT * FROM DEACORE.SALES;`
ORA-00376: file cannot be read at this time
- Но другие секции таблицы доступны



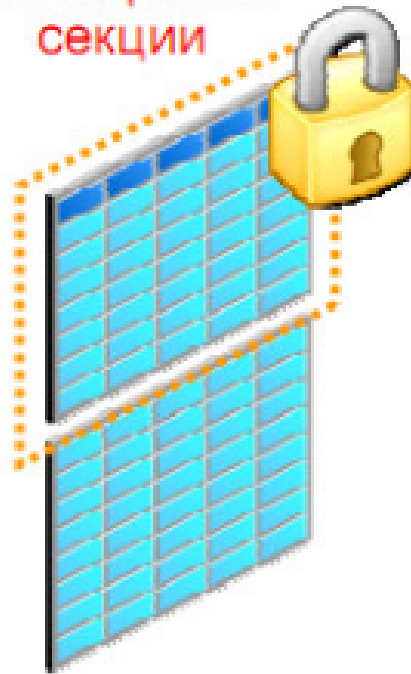
Секционирование и блокировки

Level up в иерархии блокировок

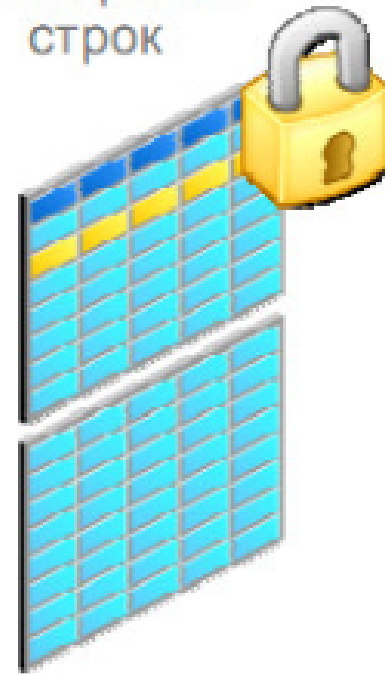
Блокировка
таблицы



Блокировка
секции



Блокировка
строк



Операции над секциями

- *ALTER TABLE RENAME PARTITION*

```
ALTER TABLE DEACore.sales RENAME PARTITION sales_11q1  
TO sales_new_name
```

- *ALTER TABLE DROP PARTITION*

```
ALTER TABLE DEACore.sales DROP PARTITION sales_11q1
```

- *ALTER TABLE MOVE PARTITION*

```
ALTER TABLE DEACore.sales MOVE PARTITION sales_11q1  
TABLESPACE data03 NOLOGGING;
```

- *ALTER TABLE ADD PARTITION*

```
ALTER TABLE DEACore.sales ADD PARTITION sales_max  
VALUES LESS THAN (MAXVALUE) TABLESPACE data00
```

Операции над секциями

- *ALTER TABLE SPLIT PARTITION*

```
ALTER TABLE DEACore.sales SPLIT PARTITION sales_11q4  
AT ('01-JUL-2012') INTO  
(PARTITION sales_sp1 TABLESPACE data01,  
PARTITION sales_sp2 TABLESPACE data02)
```

- *ALTER TABLE MERGE PARTITION*

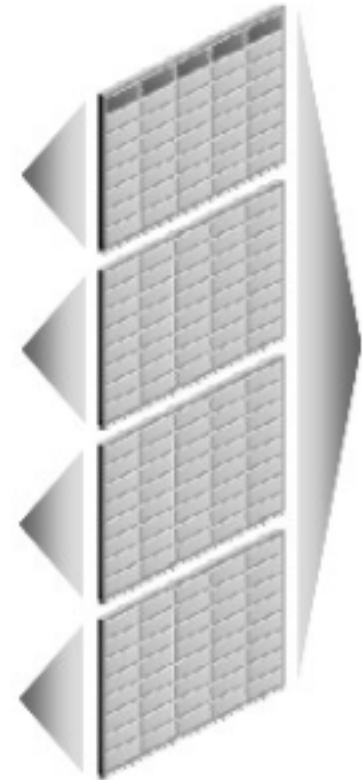
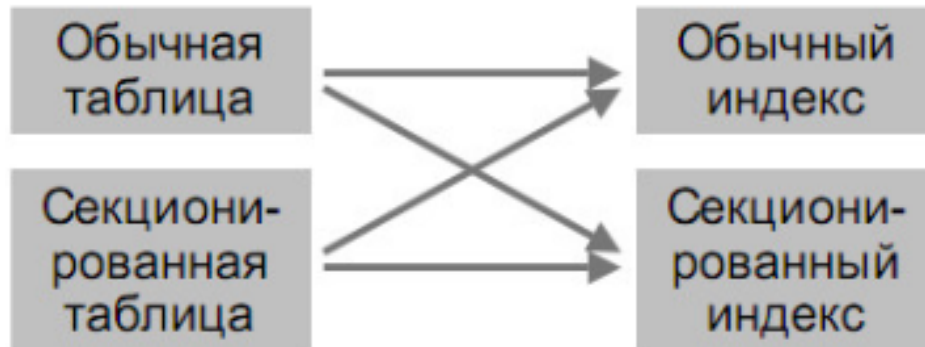
```
ALTER TABLE DEACore.sales MERGE PARTITIONS  
sales_11q1, sales_11q2 INTO sales_m1;
```

- *ALTER TABLE EXCHANGE PARTITION*

```
ALTER TABLE DEACore.sales EXCHANGE PARTITION sales_11q2  
WITH TABLE DEACore.my_sales WITHOUT VALIDATION;
```

Секционирование и индексы

- Для ускорения доступа таблицы могут индексироваться
- Индексы могут секционироваться

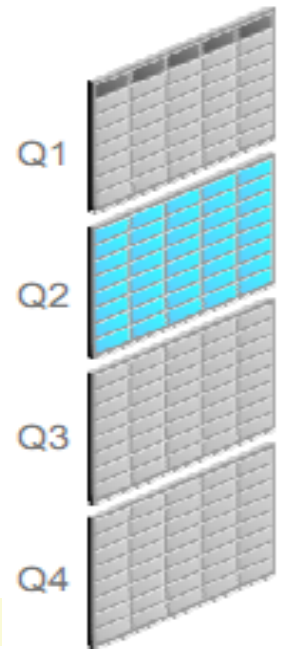


Локальные и глобальные индексы

- **Локальные**(ВСЕГДА секционированные)
 - Создается автоматически для каждой секции
- **Глобальные**(ИНОГДА секционированные)
 - Создаются и управляются вручную

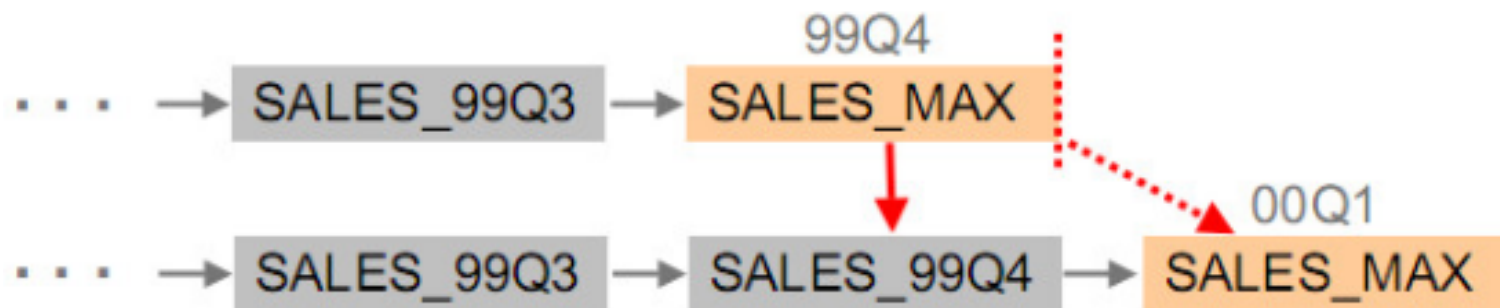
Диапазонное (*range*) секционирование

```
CREATE TABLE DEACore.sales
(
  prod_id NUMBER,
  time_id DATE
)
PARTITION BY RANGE (time_id)
(
  PARTITION sales_11q1 VALUES LESS THAN ('01-APR-2011') TABLESPACE data00,
  PARTITION sales_11q2 VALUES LESS THAN ('01-JUL-2011') TABLESPACE data01,
  PARTITION sales_11q3 VALUES LESS THAN ('01-OCT-2011') TABLESPACE data02,
  PARTITION sales_11q4 VALUES LESS THAN ('01-JAN-2012') TABLESPACE data03,
  PARTITION sales_max VALUES LESS THAN (maxvalue) TABLESPACE data04
);
```



Недостаток диапазонного секционирования

- При загрузке новых данных в таблицу нужно постоянно расщеплять секцию MAXVALUE



Интервальное секционирование

```
create table t3 (x1 nvarchar2(50), x2 number(3))  
partition by range (x2)  
interval (100) store in (users)  
(  
    partition t11 values less than (101)  
);
```

- Создается единственная диапазонная секция без MAXVALUE
- Новые секции будут создаваться **автоматически** (по 1ой операции INSERT, не попадающей в диапазоны существующих секций)

Интервальное секционирование

```
select * from user_TAB_PARTITIONS where table_name = 'T3';
```

Script Output x Query Result x						
SQL All Rows Fetched: 2 in 0,047 seconds						
	TABLE_NAME	COMPOSITE	PARTITION_NAME	SUBPARTITION_COUNT	HIGH_VALUE	
1	T3	NO	T11	0	101	
2	T3	NO	SYS_P471	0	801	

```
select * from user_TAB_PARTITIONS where table_name = 'T3';
```

Script Output x Query Result x						
SQL All Rows Fetched: 4 in 0,032 seconds						
	TABLE_NAME	COMPOSITE	PARTITION_NAME	SUBPARTITION_COUNT	HIGH_VALUE	
1	T3	NO	T11	0	101	
2	T3	NO	SYS_P472	0	301	
3	T3	NO	SYS_P473	0	501	
4	T3	NO	SYS_P471	0	801	

Интервальное секционирование

```
3 select * from t3 partition for(100);
```

Script Output x



Query Result x



SQL

All Rows Fetched: 2 in 0 seconds



X1



X2

1

Восемь

8

2

Восемьдесят

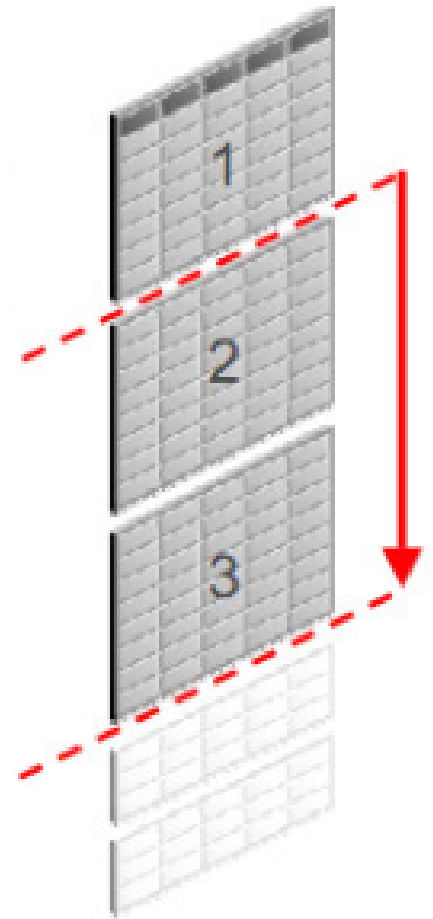
80

Изменение интервала секционирования

- Предустановленный интервал для секций можно изменить

```
alter table t3 set interval (50);
```

- На содержимом существующих интервальных секций это никак не сказывается
- Новые секции будут создаваться для интервала нового значения

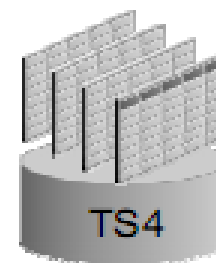
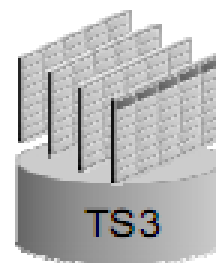
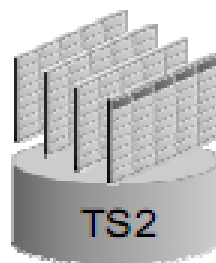
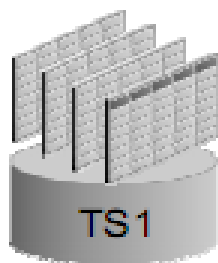


Хэш-секционирование



Хэш-секционирование

```
CREATE TABLE DEACore.salesH  
(  
  prod_id NUMBER,  
  time_id DATE  
)  
PARTITION BY HASH (prod_id)  
PARTITIONS 16  
STORE IN (data00, data01, data02, data03);
```

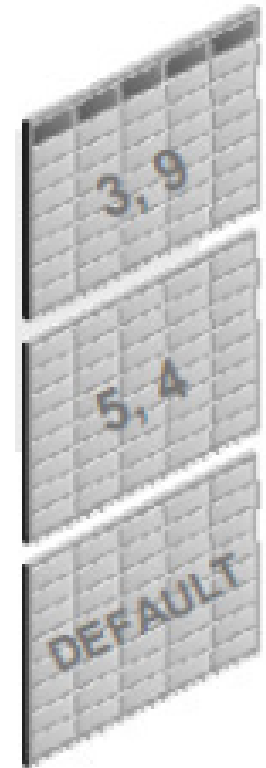


- Хэширование строк не обозначает, что строки распределятся по секциям случайным образом

Списочное секционирование

- Списочное секционирование позволяет разбить таблицу по списку дискретных значений

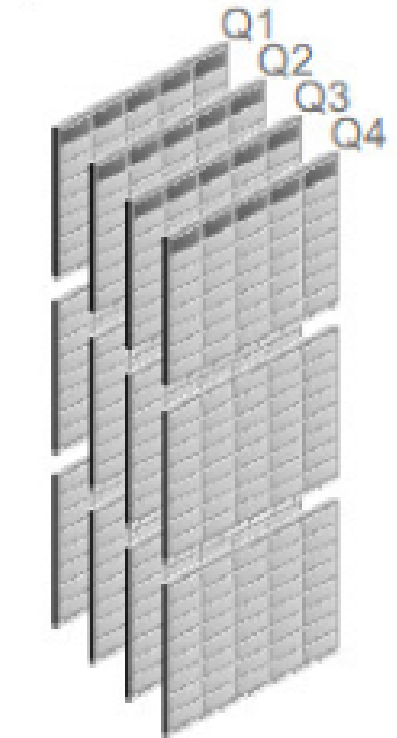
```
CREATE TABLE DEACore.salesL
(
    prod_id NUMBER,
    time_id DATE,
    group_id NUMBER
)
PARTITION BY LIST (group_id)
(
    PARTITION sales_11 VALUES (3, 9),
    PARTITION sales_12 VALUES (5, 4),
    PARTITION sales_other VALUES (DEFAULT)
);
```



Композитное секционирование

Секционирование секций

- При секционировании большой таблицы сами секции могут оказаться достаточно крупными
- Такие секции могут секционироваться по другому критерию секционирования
- Допускается 2 уровня секционирования(секции и подсекции)



Композитное секционирование

Секции и подсекции

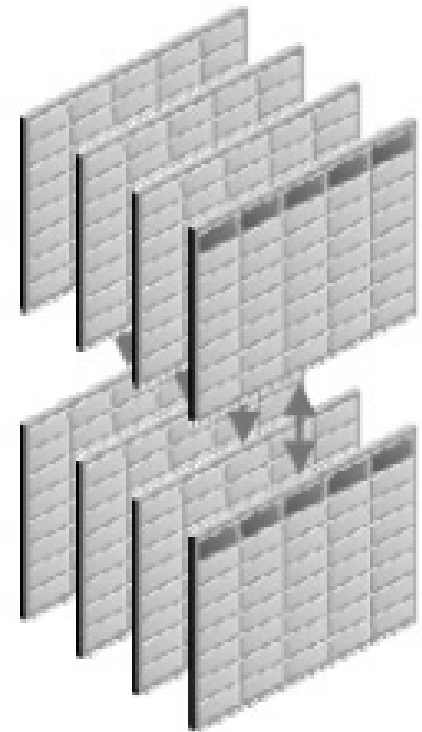
```
CREATE TABLE DEACore.salesC
(
  prod_id NUMBER,
  time_id DATE,
  cust_id NUMBER
)
PARTITION BY RANGE (time_id)
SUBPARTITION BY HASH(cust_id)
SUBPARTITIONS 4 STORE IN (data00, data01, data02, data03)
(
  PARTITION sales_1lq1 VALUES LESS THAN ('01-APR-2011'),
  PARTITION sales_1lq2 VALUES LESS THAN ('01-JUL-2011')
);
```



Эквисекционирование

Equi-partitioning

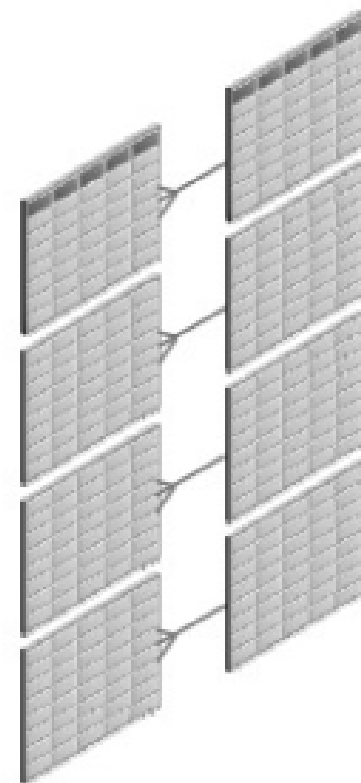
- Объекты эквисекционированы, если они секционированы одинаковым образом
- Общий метод секционирования
- Равное количество секций
- Те же границы, количество, порядок и типы столбцов в ключе
- Объекты могут быть эквисекционированы на одном или разных уровнях(секции, подсекции)



Секционирование по ссылке

Reference partitioning

- Выполняется если таблицы явным образом связаны ссылочными ограничениями целостности
- Метод и ключ секционирования подчиненной таблицы наследуется из главной
- Манипуляции с секциями главной таблицы автоматически отражаются на секциях подчиненной



Пример ссылочного секционирования

```
CREATE TABLE DEACore.salesR
(
  prod_id NUMBER,
  time_id DATE,
  fact_id NUMBER CONSTRAINT sales_pk PRIMARY KEY
)
PARTITION BY RANGE (time_id)
(
  PARTITION sales_llq1 VALUES LESS THAN ('01-APR-2011'),
  PARTITION sales_llq2 VALUES LESS THAN ('01-JUL-2011'),
  PARTITION sales_max VALUES LESS THAN (MAXVALUE)
);

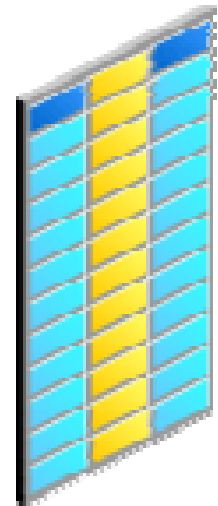
CREATE TABLE DEACore.salesR_details
(
  details_id NUMBER,
  fact_id NUMBER NOT NULL
  CONSTRAINT sales_details_fk REFERENCES DEACore.salesR(fact_id)
)
PARTITION BY REFERENCE (sales_details_fk);
```

- Родительская таблица не может быть интервальной

Виртуальные столбцы в таблице

- Значения в столбцах вычисляются динамически(“на лету”)

```
CREATE TABLE DEACore.salesR
(
  prod_id NUMBER,
  AverageRevenue NUMBER(10, 2),
  TotalProduct NUMBER(10, 2),
  TotalRevenue AS (AverageRevenue*TotalProduct)
);
```



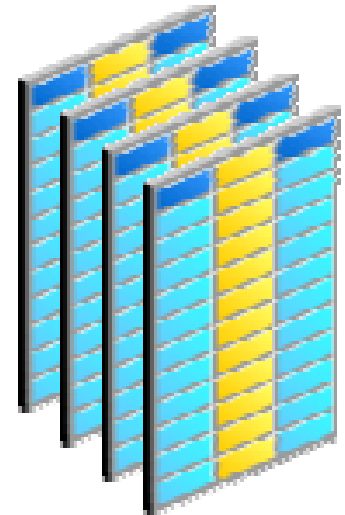
- Виртуальными могут быть только столбцы в обычных таблицах

Описание виртуального столбца

- Выражение должно быть детерминистическим (независимым от времени)
- Выражение может ссылаться только на другие столбцы из этой же таблицы(но не виртуальные)

Секционирование по виртуальному столбцу

```
CREATE TABLE DEACore.salesR
(
  prod_id NUMBER,
  AverageRevenue NUMBER(10, 2),
  TotalProduct NUMBER(10, 2),
  TotalRevenue AS (AverageRevenue*TotalProduct)
)
PARTITION BY RANGE (TotalRevenue)
(
  PARTITION TR1 VALUES LESS THAN (1000) TABLESPACE data00,
  PARTITION TR2 VALUES LESS THAN (5000) TABLESPACE data01,
  PARTITION TR3 VALUES LESS THAN (10000) TABLESPACE data02,
  PARTITION TR_max VALUES LESS THAN (maxvalue) TABLESPACE data03
);
```



- Виртуальный столбец может быть использован практически в любых схемах секционирования

Зачем секционировать данные?

- Секционирование повышает производительность обработки данных
- Отбрасывание секций, параллелизм, балансировка дисковой нагрузки
- Секционирование упрощает управление крупными объектами хранения
- Крупный объект разбивается на индивидуально-управляемые части
- Секционирование обеспечивает дополнительную надежность системы
- Сокращение времени обслуживания, нечувствительность к сбоям