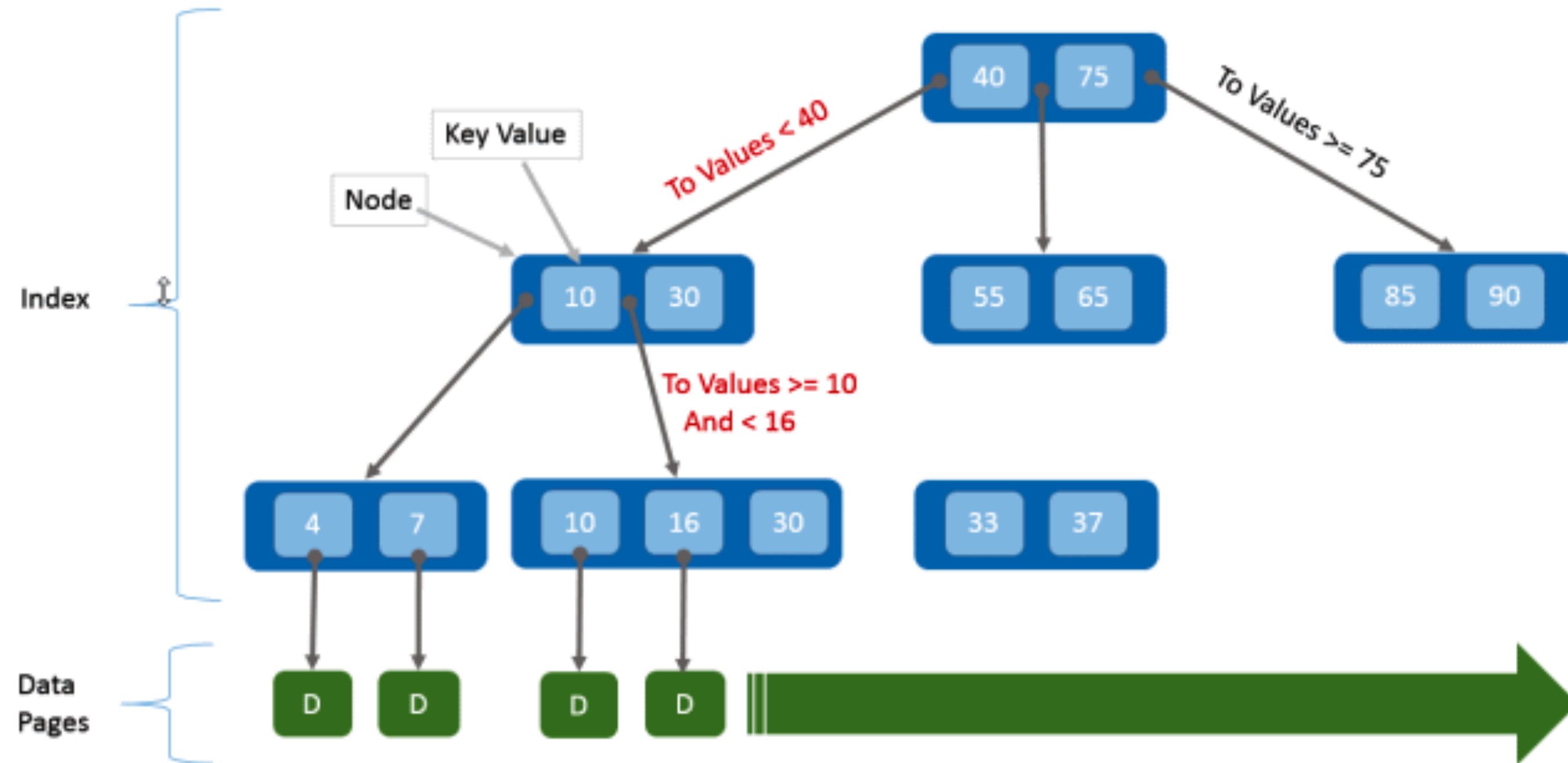


**Индексы, партиционирование
и сегментирование.**

Индексы

B-Tree Layout



Индекс – это объект базы данных, обеспечивающий дополнительные способы быстрого поиска и извлечения данных.

Плюсы:

- При наличии индекса время поиска нужных строк можно существенно уменьшить.

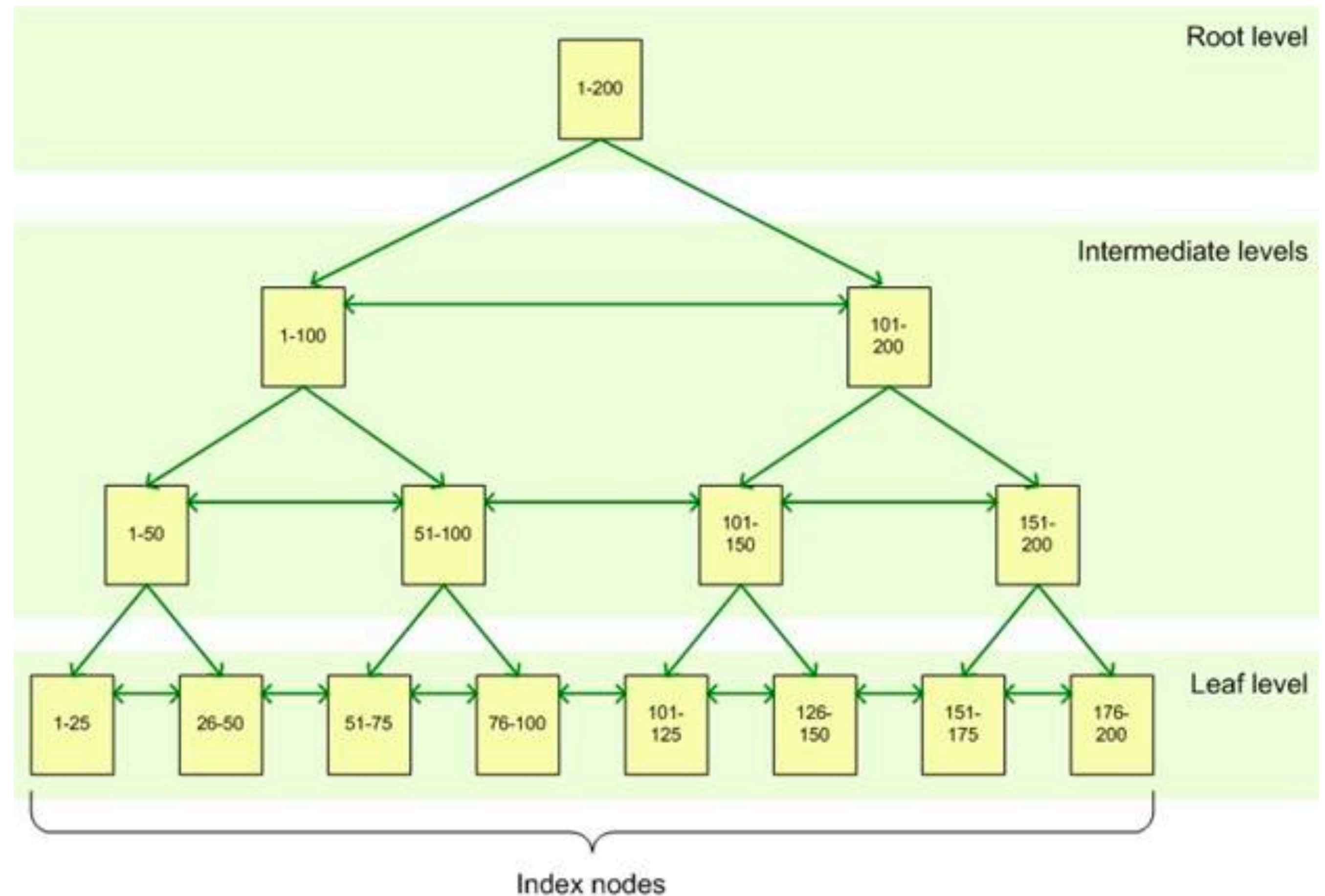
Минусы:

- Дополнительное место на диске и в оперативной памяти,
- Замедляются операции вставки, обновления и удаления записей.

Классификация индексов

Существует два типа индексов:

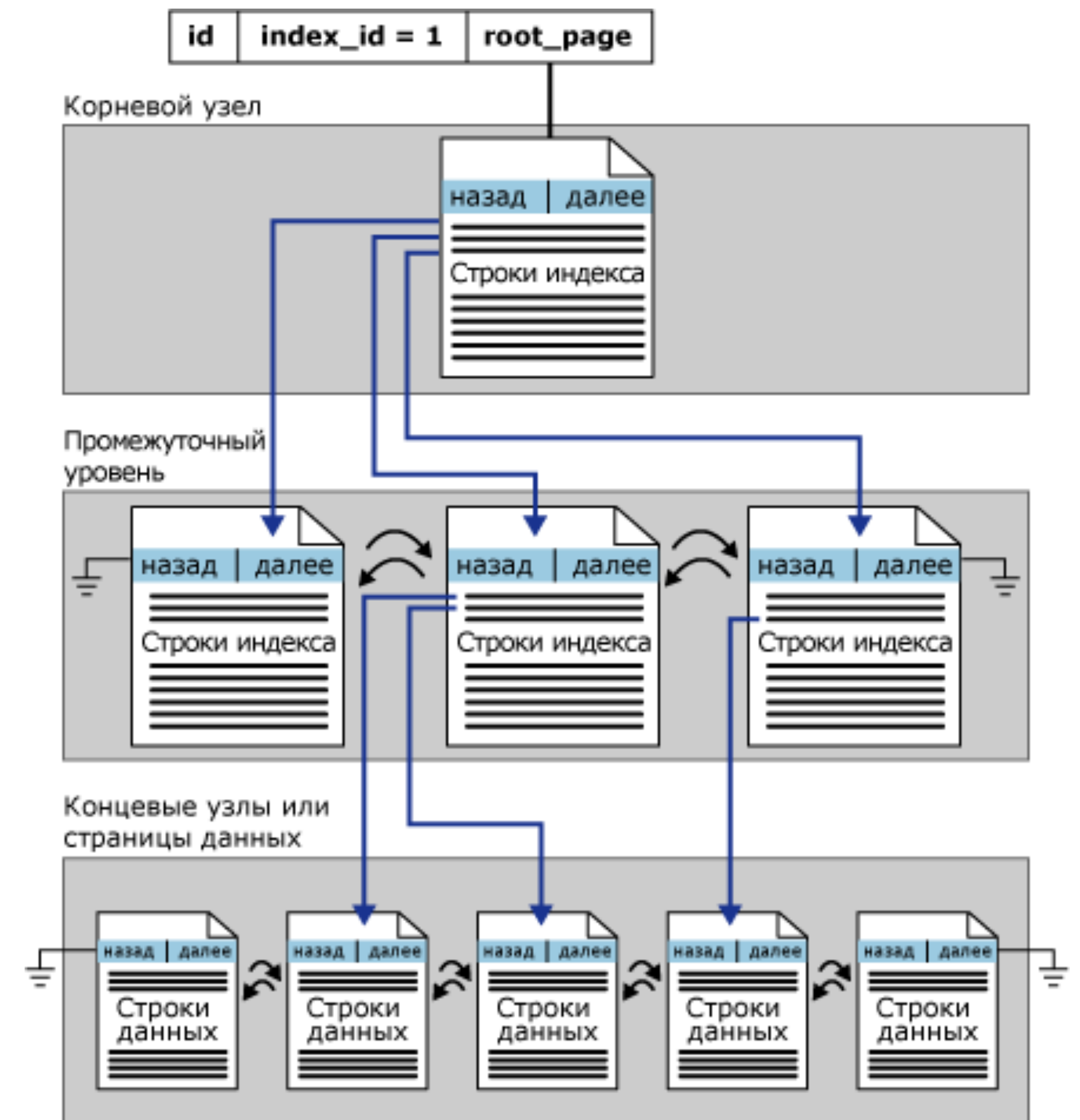
- Кластерные индексы;
- Некластерные индексы, которые включают:
 - на основе кучи;
 - на основе кластерных индексов.



Кластерные индексы

В кластерном индексе таблица представляет собой часть индекса, или индекс представляет собой часть таблицы.

- Листовой узел кластерного индекса – это страница таблицы с данными.
- Поскольку сами данные таблицы являются частью индекса, то очевидно, что для таблицы может быть создан только один кластерный индекс.
- Кластерный индекс является уникальным индексом по определению. Это означает, что все ключи записей должны быть уникальные.



Пример создания индекса

CREATE [UNIQUE] [CLUSTERED | NONCLUSTERED]
INDEX имя-индекса
ON имя-таблицы-или-представления
(список-столбцов)
[INCLUDE (список-столбцов)]
[WITH список-опций]
[ON файловая-группа]

— Создание индекса отдельно от таблицы

```
CREATE UNIQUE NONCLUSTERED INDEX [IX_Address]  
ON [Person].[Address] ([AddressLine1] ASC,  
                        [AddressLine2] ASC,  
                        [City] ASC,  
                        [StateProvinceID] ASC,  
                        [PostalCode] ASC)
```

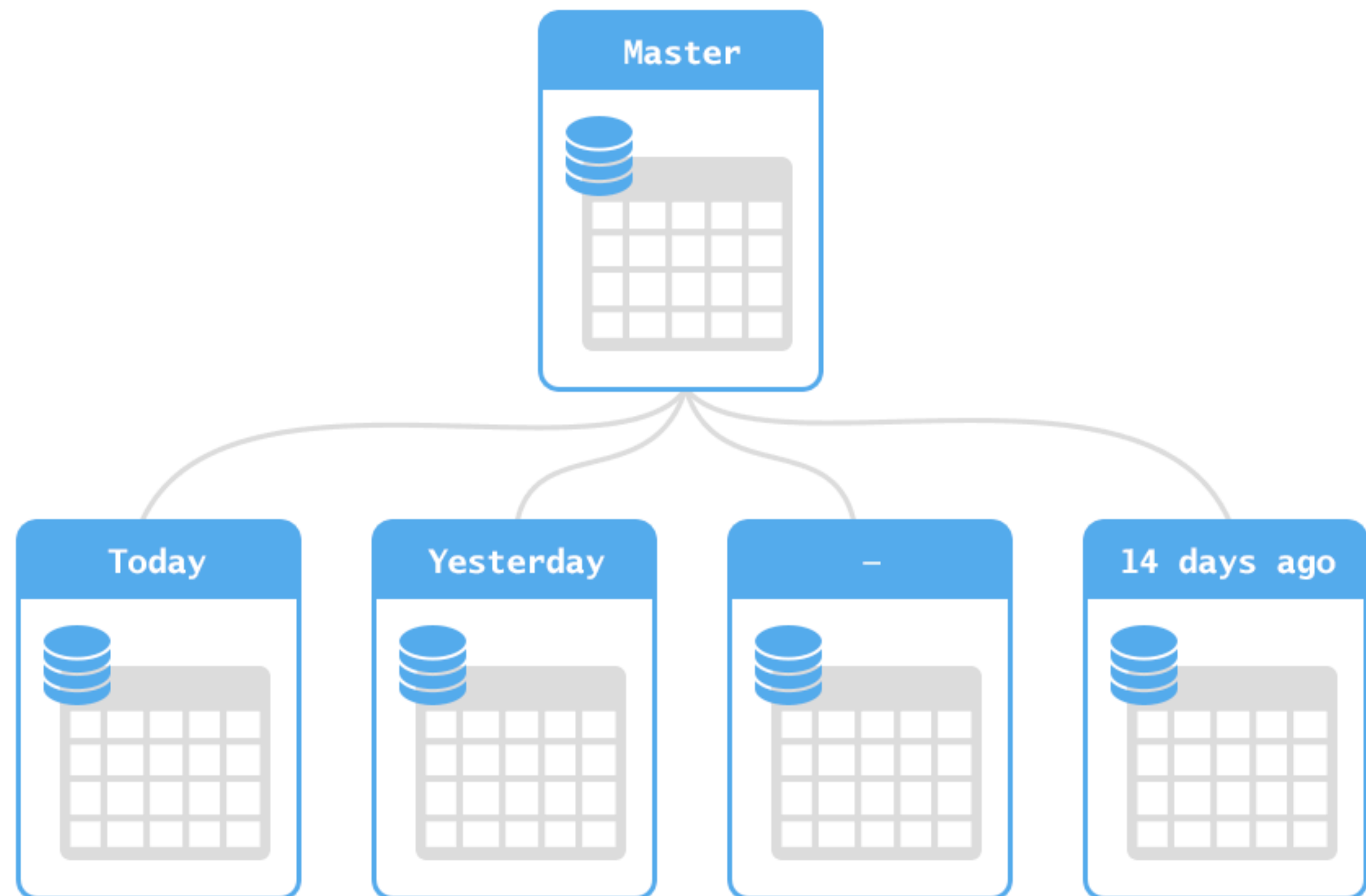
— Создание индекса вместе с ограничением

```
CREATE TABLE [Person].[Address](  
    [AddressID] int IDENTITY(1,1) NOT NULL,  
    ...  
    CONSTRAINT [PK_Address_AddressID] PRIMARY KEY  
        CLUSTERED ([AddressID] ASC) ON [PRIMARY]  
) ON [PRIMARY]
```


Партиционирование

Партиционирование – это метод разделения больших (исходя из количества записей, а не столбцов) таблиц на много маленьких. И желательно, чтобы это происходило прозрачным для приложения способом.

```
CREATE TABLE measurement (  
  city_id      int not null,  
  logdate     date not null,  
  peaktemp    int,  
  unitsales   int  
) PARTITION BY RANGE (logdate);
```



Сегментирование

Для распределения данных по кластерам используется их сегментация(Segmentation), а точнее сегментация проекций(Projection) в которых они находятся. Задача разработчика — подобрать такой список полей и/или такую функцию(например, хэш-функцию), благодаря которым данные равномерно распределятся по нодам кластера.

Создадим простую таблицу:

```
CREATE TABLE test_order (  
  n integer primary key,  
  attr1 text,  
  attr2 text,  
  dt timestamp  
)  
Distributed by (n)  
Partition by dt;
```

Создадим две сообщённые проекции:

```
CREATE PROJECTION p_test_b0 AS  
SELECT n FROM test_order BY n  
SEGMENTED BY hash(n) ALL NODES;
```

```
CREATE PROJECTION p_test_b1 AS  
SELECT n, attr1 FROM test_order BY n  
SEGMENTED BY hash(n) ALL NODES offset 1;
```

3 Segmented across 4 nodes

Node 1

| order_id | order_date | item | qty |
|----------|------------|------|-----|
| 1 | 01/02/06 | A | 1 |
| 5 | 12/22/06 | D | 1 |
| 9 | 05/06/07 | A | 1 |
| 13 | 09/18/08 | E | 2 |

Node 2

| order_id | order_date | item | qty |
|----------|------------|------|-----|
| 2 | 02/04/07 | B | 1 |
| 6 | 01/02/08 | A | 1 |
| 10 | 12/22/06 | B | 1 |
| 14 | 05/06/07 | B | 1 |

Node 3

| order_id | order_date | item | qty |
|----------|------------|------|-----|
| 3 | 09/18/08 | A | 1 |
| 7 | 02/04/07 | B | 10 |
| 11 | 01/02/06 | C | 1 |
| 15 | 12/22/06 | D | 1 |

Node 4

| order_id | order_date | item | qty |
|----------|------------|------|-----|
| 4 | 05/06/07 | C | 3 |
| 8 | 09/18/08 | E | 4 |
| 12 | 02/04/07 | A | 1 |
| 16 | 01/02/06 | B | 1 |

4 Partitioned by year on single a node

Node 2

| | order_id | order_date | Item | qty |
|------|----------|------------|------|-----|
| 2006 | 10 | 12/26/06 | B | 1 |
| 2007 | 2 | 02/04/07 | B | 1 |
| | 14 | 05/06/07 | B | 1 |
| 2008 | 6 | 01/02/08 | A | 1 |

1 Table data

| order_id | order_date | item | qty | HASH(order_id) |
|----------|------------|------|-----|----------------|
| 1 | 01/02/06 | A | 1 | 1 |
| 2 | 02/04/07 | B | 1 | 2 |
| 3 | 09/18/08 | A | 1 | 3 |
| 4 | 05/06/07 | C | 3 | 4 |
| 5 | 12/22/06 | D | 1 | 1 |
| 6 | 01/02/08 | A | 1 | 2 |
| 7 | 02/04/07 | B | 10 | 3 |
| 8 | 09/18/08 | E | 4 | 4 |
| 9 | 05/06/07 | A | 1 | 1 |
| 10 | 12/22/06 | B | 1 | 2 |
| 11 | 01/02/06 | C | 1 | 3 |
| 12 | 02/04/07 | A | 1 | 4 |
| 13 | 09/18/08 | E | 2 | 1 |
| 14 | 05/06/07 | B | 1 | 2 |
| 15 | 12/22/06 | D | 1 | 3 |
| 16 | 01/02/06 | B | 1 | 4 |

2 Segmented by HASH