



Онлайн образование



Проверить, идет ли запись

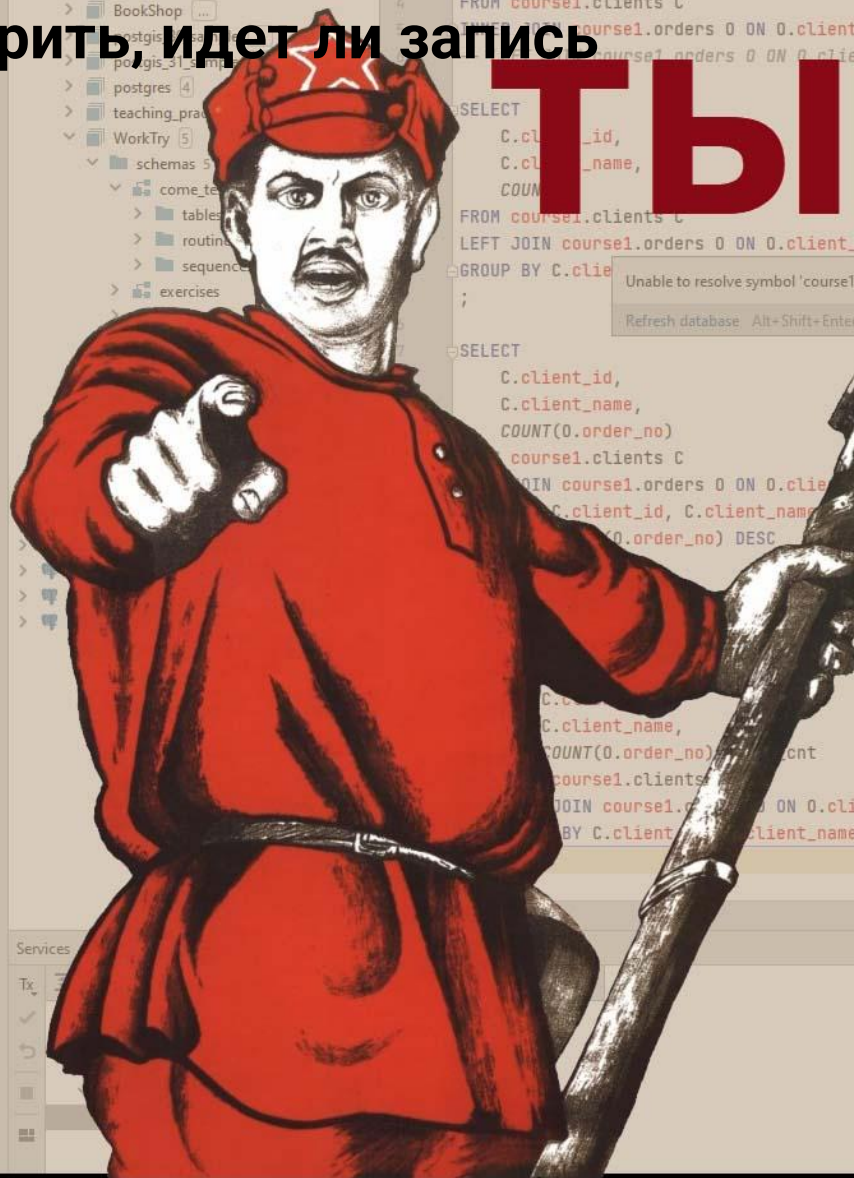
Меня хорошо видно && слышно?



• REC

Проверить, идет ли запись

ТЫ



ВКЛЮЧИЛ ЗАПИСЬ?

Тема вебинара

Секционирование таблиц в PostgreSQL



Золотов Антон Владимирович

Преподаватель



Антон Золотов

Программист, разработчик баз данных.
Общий стаж разработки программного обеспечения
более 35 лет, разработкой БД занимаюсь около 20 лет.

Предметные области:

- ERP;
 - складская логистика;
 - системы анализа медиаданных;
 - системы поддержки принятия решений.
-
- сертифицированный тренер PostgreSQL
 - преподаватель курса "PostgreSQL" в OTUS

Правила вебинара



Активно
участвуем



Off-topic обсуждаем
в Telegram



Задаем вопрос
в чат или голосом



Вопросы вижу в чате,
могу ответить не сразу

Условные обозначения



Индивидуально



Время, необходимое
на активность



Пишем в чат



Говорим голосом

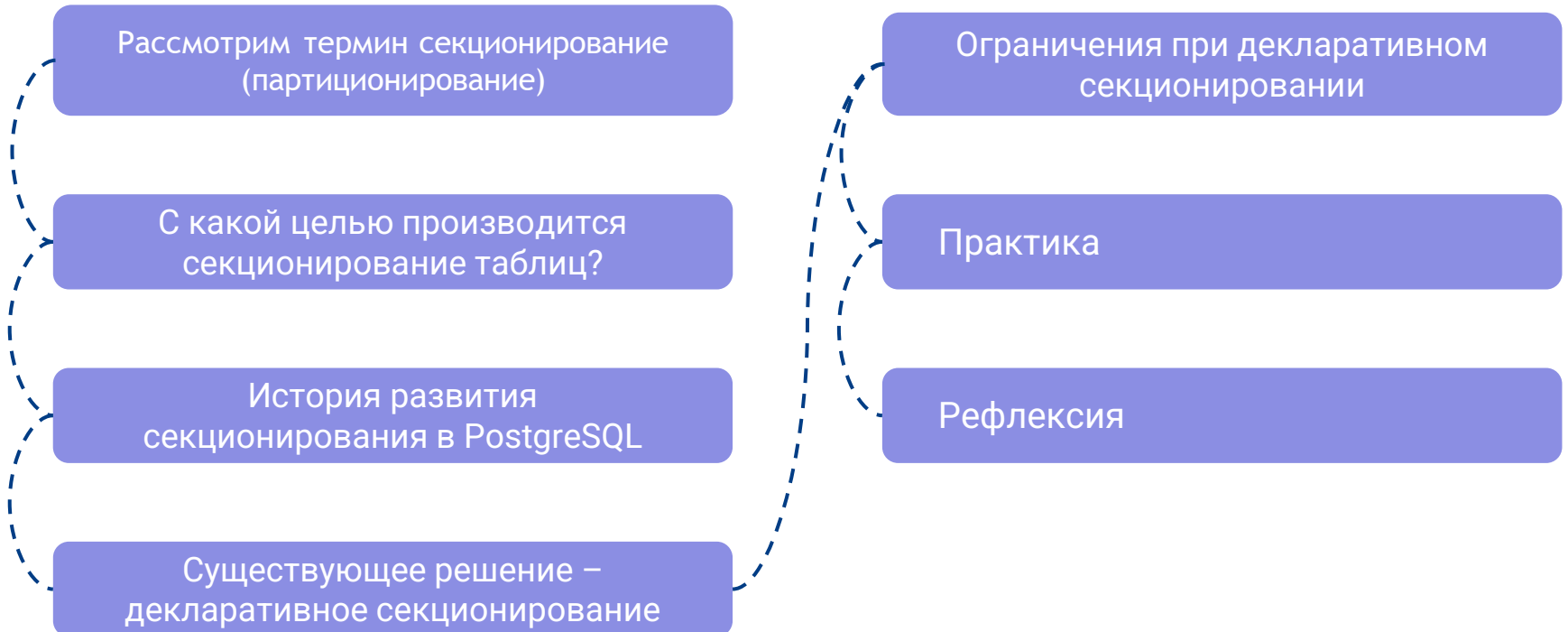


Документ



Ответьте себе или
задайте вопрос

Маршрут вебинара



Цели вебинара

После занятия вы сможете

1. Создавать секционированные таблицы
2. Обслуживать секционированные таблицы
3. Увеличить производительность запросов

Смысл

Зачем вам это уметь

1. Для лучшего хранения большого количества данных
2. Для более быстрого доступа к данным
3. Для ускорения запросов

Секционирование таблиц в PostgreSQL

Секционирование.

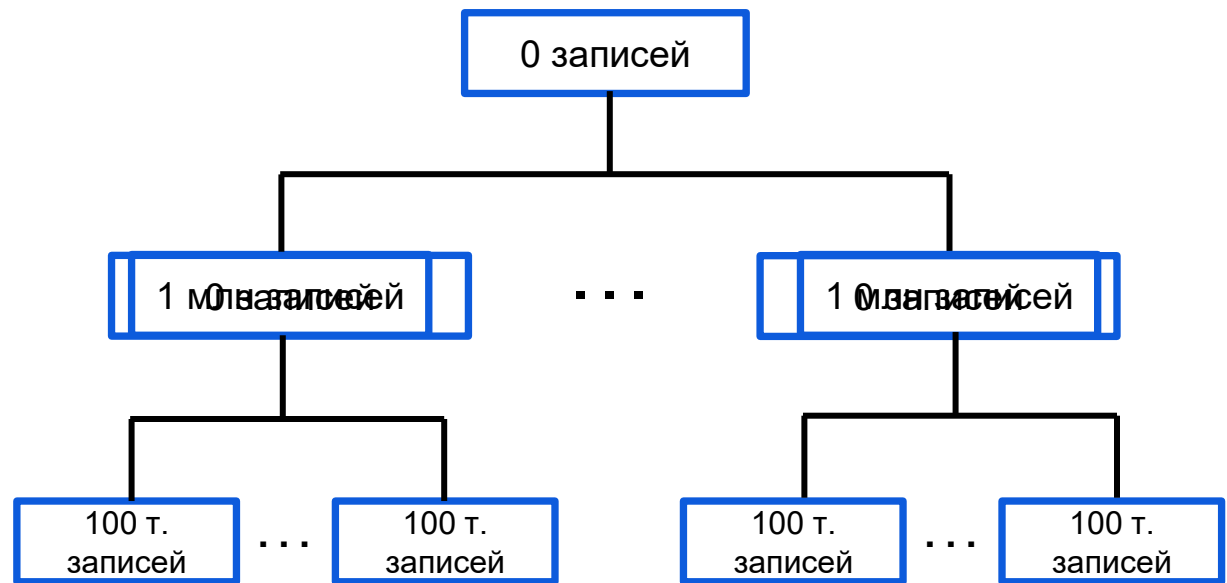
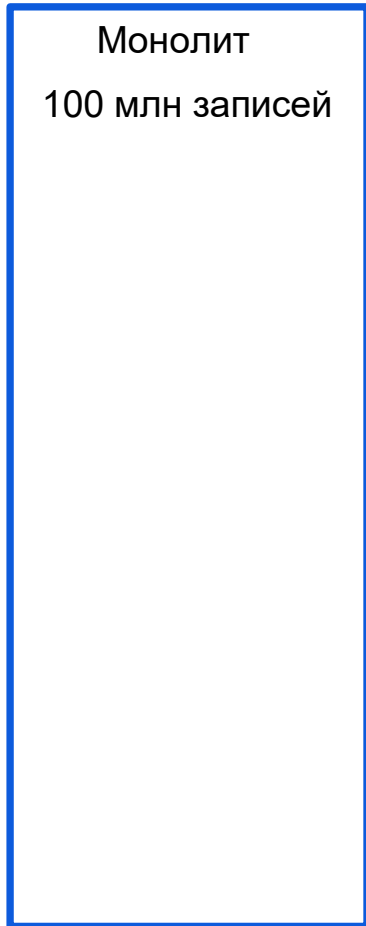
Цели, задачи

Партиционирование (секционирование) - разбиение больших таблиц на относительно небольшие части (партиции, секции).

Возможные выгоды секционирования:

- увеличение производительности запросов в тех случаях, когда значительная доля запрашиваемых строк принадлежит одной (в идеальном случае) или небольшому числу секций;
 - за счет того, что поиск записей производится в таблице меньшего размера;
 - в случае, когда в выборке задействована большая часть записей одной секции планировщик может выбрать последовательное сканирование секции, а не поиск по индексу и, как следствие, произвольное (неупорядоченное) чтение данные разбросанных по большой таблице;
- массовая загрузка и массовое удаление данных могут осуществляться посредством добавления и удаления секций.

Секционирование



Данные содержатся только в таблицах-листьях дерева секций

История: секционирование с использованием механизма наследования таблиц

Секционирование наследованием (до 10-й версии PG)

- Создаётся таблица и от неё наследуются секции с помощью команды:
CREATE TABLE table_s1 () INHERITS (table);
- В дочерних таблицах создаются ограничения, которые не пересекаются между секциями **CHECK (date between...)**
- Создаётся триггер, который распределяет строки
**CREATE TRIGGER trigger_name
BEFORE INSERT ON table_name
FOR EACH ROW EXECUTE PROCEDURE func_name();**

Декларативное секционирование

Декларативный способ

- Относительно новый способ (с версии 10)
- Проще переносить (т.к. синтаксис теперь похож на другие СУБД)
- Проще реализовать (без триггеров)
- Можно создавать подсекции

*Не забыть включить (точнее – не выключать) **enable_partition_pruning** для оптимизации запросов*

Секционирование

Общий синтаксис для секционирования таблицы:

CREATE TABLE ...

(

...

)

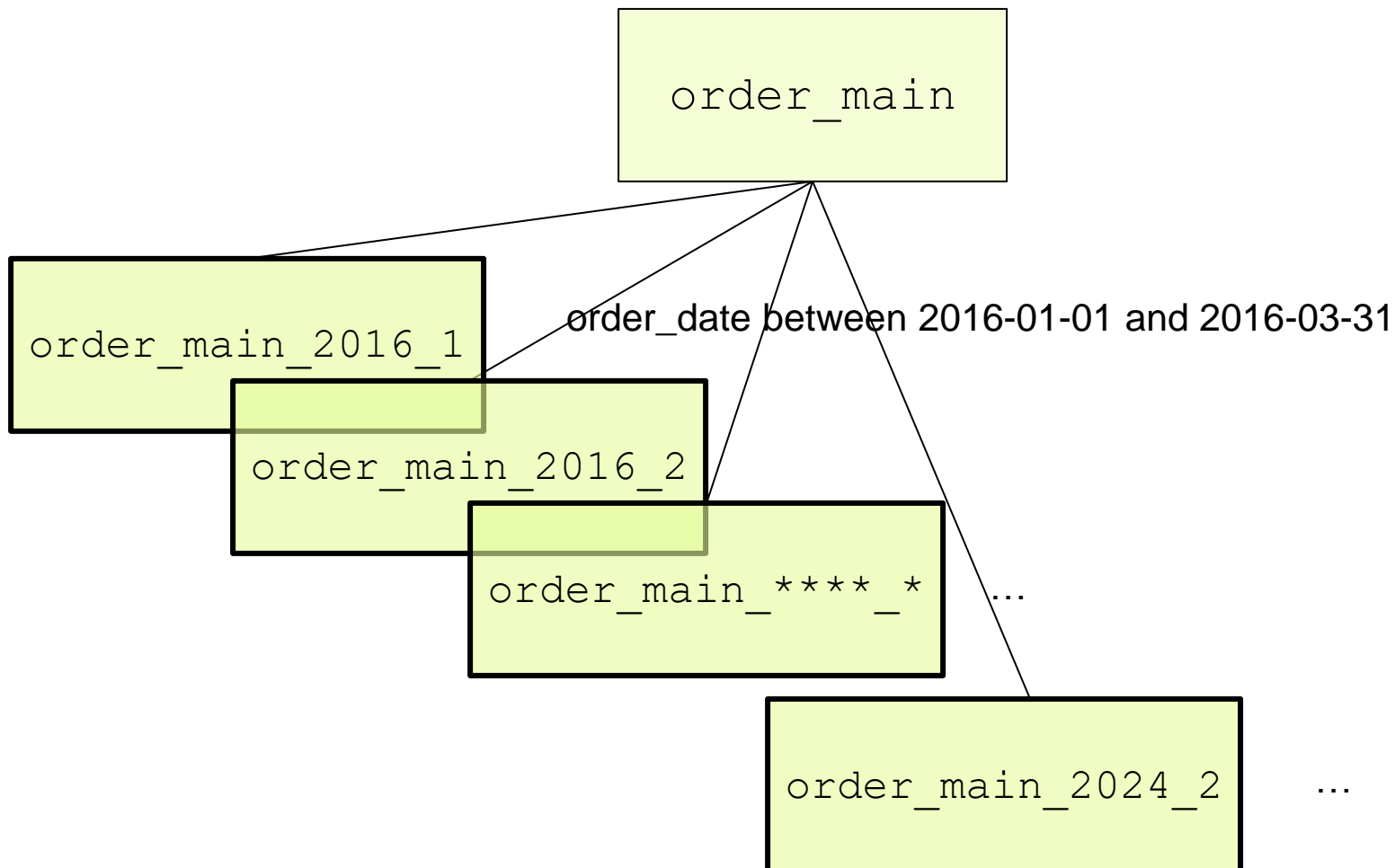
PARTITION BY { RANGE | LIST | HASH } ({ имя_столбца | (выражение) }

[COLLATE правило_сортировки] [класс_операторов] [, ...])

CREATE TABLE section_name PARTITION OF table_name FOR VALUES FROM ('2006-02-01') TO ('2006-03-01');

Удаление партиции – DROP TABLE section_name;

Секционирование по диапазону значений



Каждая секция тоже может быть секционирована

Секционирование по диапазону значений

Синтаксис:

Таблица

```
CREATE TABLE table_name
(
...
)
PARTITION BY RANGE ({ имя_столбца | ( выражение )
                    [ COLLATE правило_сортировки ]
                    [ класс_операторов ] [, ... ]})
```

Секция

```
CREATE TABLE section_name PARTITION OF table_name
    FOR VALUES FROM (***) TO (***);
```

```
CREATE TABLE section_name PARTITION OF table_name DEFAULT;
```

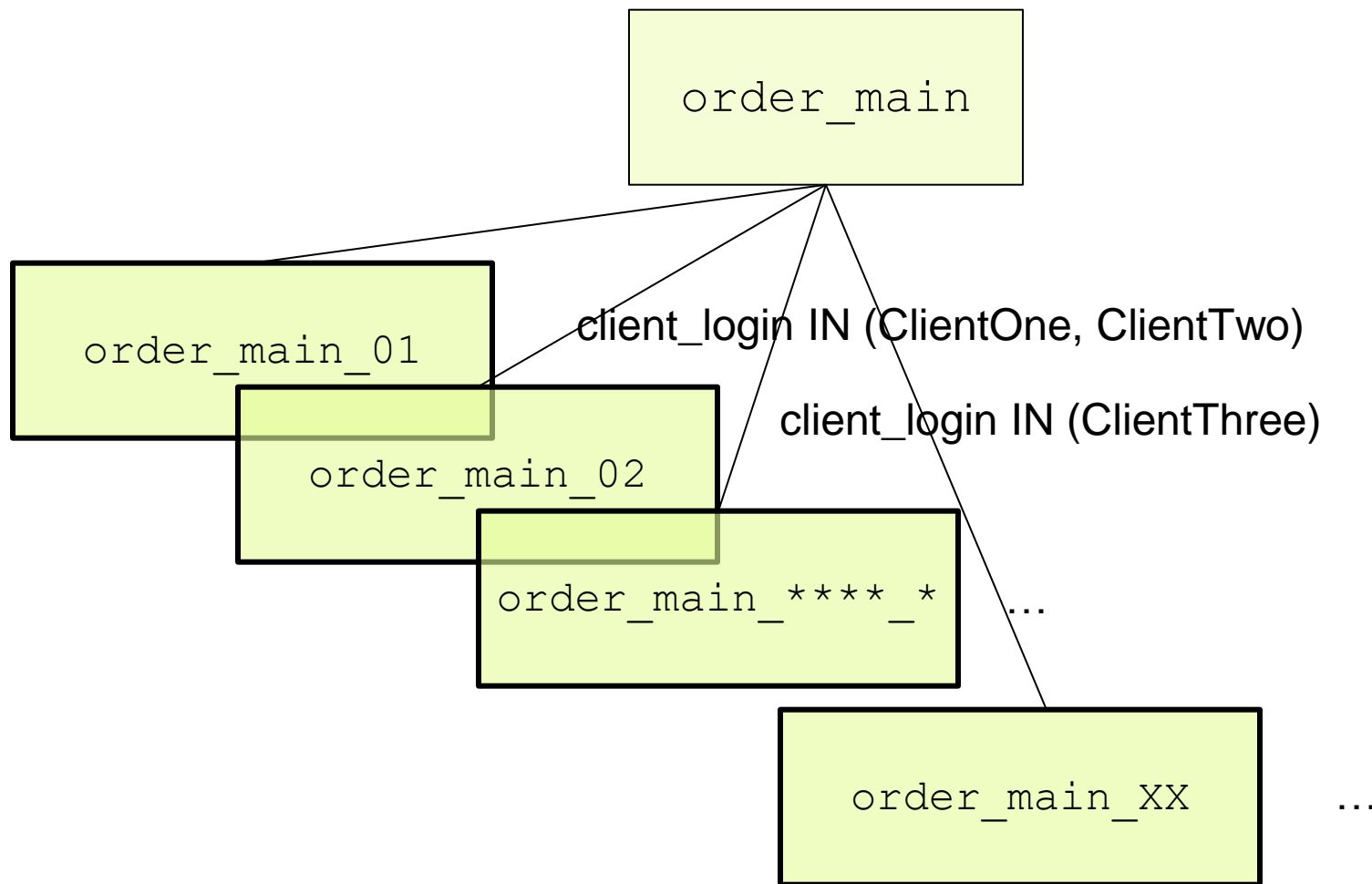
Возможно использование MINVALUE и MAXVALUE

Секционирование по диапазону значений

Как создать новую секцию для текущего года/квартала/месяца...?

- Триггер?
- Интерфейсная функция?
- Задача CRON?

Секционирования по списку значений



Каждая секция тоже может быть секционирована

Секционирования по списку значений

Синтаксис:

Таблица

```
CREATE TABLE  
(  
...  
)  
PARTITION BY LIST ({ имя_столбца | ( выражение )  
    [ COLLATE правило_сортировки ]  
    [ класс_операторов ] [, ... ]})
```

Секция

...

```
CREATE TABLE section_name PARTITION OF table_name  
    FOR VALUES IN (***, ...);
```

Возможно создание секции по умолчанию

Секционирования по хешу

Нужно равномерное распределение (требуется поле с уникальными или почти уникальными значениями)

Нет явного кандидата в ключи разбиения

Секционирования по хешу

Синтаксис:

Таблица

```
CREATE TABLE  
(  
...  
)  
PARTITION BY HASH ({ имя_столбца | ( выражение )  
    [ COLLATE правило_сортировки ]  
    [ класс_операторов ] [, ... ]})
```

Секция

...

```
CREATE TABLE section_name PARTITION OF table_name  
FOR VALUES WITH (MODULUS n, REMAINDER k); ...
```


Ограничения

*...и грабли те же самые приветствуют меня...
В.Вишневский*

- Нельзя секционировать уже созданную таблицу
- Все секции должны иметь тот же набор столбцов, что и «головная» секционируемая таблица¹
- Все столбцы, по значениям которых производится секционирование («ключ секционирования» или «ключ разбиения» должны входить в ограничения уникальности (включая уникальные индексы и РК)
- При секционировании BY RANGE ключ секционирования не может принимать NULL-значения

¹При создании секции, как «PARTITION OF» условие выполняется автоматически, но при присоединении («ATTACH») выполнение этого условия ложится на разработчика

Присоединение и отсоединение секций

Секции таблицы можно отсоединять (командой **DETACH**) и присоединять (командой **ATTACH**)

Прежде чем выполнять команду **ATTACH PARTITION**, рекомендуется создать ограничение **CHECK** в присоединяемой таблице, соответствующее ожидаемому ограничению секции, как показано выше. Благодаря этому система сможет обойтись без сканирования, необходимого для проверки неявного ограничения секции. Без этого ограничения **CHECK** нужно будет просканировать и убедиться в выполнении ограничения секции, удерживая блокировку **ACCESS EXCLUSIVE** в этой секции. После выполнения команды **ATTACH PARTITION** рекомендуется удалить ограничение **CHECK**, поскольку оно больше не нужно. Если присоединяемая таблица также является секционированной таблицей, то каждая из её секций будет рекурсивно блокироваться и сканироваться до тех пор, пока не встретится подходящее ограничение **CHECK** или не будут достигнуты конечные разделы.

Вопросы?



Ставим “+”,
если вопросы есть



Ставим “-”,
если вопросов нет

Рефлексия