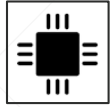




Методы использования Sequences в бизнес-логике

Камиль Исламов

План



- Общие сведения о Sequences



- Возможности Sequences у PostgreSQL



- Варианты использования Sequences в хранимых процедурах



- Реализация бизнес-логики через глобальные последовательности



- Итоги



Некоторые способы обеспечения уникальности данных



- Автоинкремент SERIAL (заменяется PostgreSQL на sequence)
- Вручную созданные Sequence отдельно для таблицы



- Различные версии UUID с ограничителями по факторам – может потребоваться соответствующий extension



- Собственно написанные функции, генерирующие идентификатор по ситуации



- Получение Max(ID) из текущих значений таблицы
- Другие способы (random, системное время до наносекунд, и т.д.)



Некоторые преимущества типовых Sequence



- Index-оптимальный формат данных bigint



- Информация о взаимной последовательности создания объектов



- Принципиально схожий механизм использования в разных СУБД



- Наименее ресурсоёмкий способ получения уникального значений



- Единый источник уникального значения для автоинкремента и «ручного»

- Отработанные способы получения данных из клиента СУБД



Некоторые преимущества альтернативных способов получения идентификаторов



- Выигрыш производительности для сверх-большого объёма данных



- Отсутствие обращения к БД для многократного применения в пределах одной транзакции



- Возможность вычисления идентификатора без обращения к БД



- Отсутствие методов взаимодействия с различными СУБД



- Избежание конфликтов идентификаторов при мультимастер-архитектуре



Два вида значений Sequence



- *Next Value* – вычисление следующего значения Sequence



- + инициация текущего значения в сессии



- *Current Value* – текущее значение



- ! Доступно в сессии только после инициации и получения значения



Дополнительный функционал в PostgreSQL



- Синтаксис вызова как *Функции*: `nextval`, `currval`, `setval`



- Update функционал – возможность явно указать значение – `setval`



- Возможность подать переменную в наименование Sequence



- Возможность динамически выбирать ту или иную Sequence в бизнес-логике через переменную



- SELECT каждой функции возвращает текущее значение Sequence после вызова `***val`



- Полноценная использование в *On Conflict* функционале



Альтернативные варианты применения Sequence



- Вычисленный nextval в переменной для переиспользования
- Значение Sequence как номер бизнес-элемента
- Sequence как источник единых идентификаторов разных таблиц
- Sequence как единая бизнес-последовательность разных сущностей
- Sequence как объединяющий несколько сущностей идентификатор
- Sequence как часть сгенерированного текстового идентификатора
- Sequence как дробная часть временной уникальной последовательности



Вычисленный nextval в переменной для переиспользования в функциях

f_x

- Удобный способ заблаговременно использовать «будущий» ID
 - Подготовить полный комплект данных для SQL запросов
 - Возможность широко использовать как переменную (массивы, параметры, ...)
 - Использование в CTE запросах как виртуальные столбцы

■ ■ ■
■ ■ ■
■ ■ ■

- Оптимизация и читаемость кода хранимых функций
 - Одна и та же переменная в нескольких SQL запросах
 - Использование переменной в массовых вычислениях
 - Отсутствие ограничений от имени Sequence в нейминге



Вычисленный nextval в переменной для переиспользования в функциях

f_x

- Удобный способ заблаговременно использовать «будущий» ID
 - Подготовить полный комплект данных для SQL запросов
 - Возможность широко использовать как переменную (массивы, параметры, ...)
 - Использование в CTE запросах как виртуальные столбцы



- Оптимизация и читаемость кода хранимых функций
 - Одна и та же переменная в нескольких SQL запросах
 - Использование переменной в массовых вычислениях
 - Отсутствие ограничений от имени Sequence в нейминге



Значение Sequence как номер бизнес-элемента

f_x

- Nextval как нумерация с возможностью пропусков
 - Не требуется вычислять следующее значение
 - Подходит для массовой генерации и импорта данных
 - Эффективно для конкурентного ввода данных



- Другие опции Sequence для бизнес-логики
 - Инкремент, который перезапускается по достижению порога CYCLE
 - Ручной сброс Sequence Setval(name,1) при необходимости (годовой № док.)
 - Гарантированные кратные значения посредством INCREMENT (чёт/нечёт, через интервал, интервал после сдвига, и т.д.)



Значение Sequence как номер бизнес-элемента

f_x

- Nextval как нумерация с возможностью пропусков
 - Не требуется вычислять следующее значение
 - Подходит для массовой генерации и импорта данных
 - Эффективно для конкурентного ввода данных



- Другие опции Sequence для бизнес-логики
 - Инкремент, который перезапускается по достижению порога CYCLE
 - Ручной сброс Sequence Setval(name,1) при необходимости (годовой № док.)
 - Гарантированные кратные значения посредством INCREMENT (чёт/нечёт, через интервал, интервал после сдвига, и т.д.)



Sequence как единая бизнес-последовательность разных сущностей

f_x

- Один и тот же Sequence как Primary Key Разных таблиц
 - Множество возможностей для полнофункциональной бизнес-логики
 - Гарантия непересекающихся ID для всех критичных бизнес-объектов
 - Журналирование всех таблиц в едином пространстве



- Преимущества единого инкремента для бизнес-логики
 - Глобальная ориентировочная шкала времени и размера всей системы
 - Возможность локализации элемента данных без названия таблицы
 - Анализ «скорости» роста инкремента как дополнительная опция мониторинга



Sequence как единая бизнес-последовательность разных сущностей

f_x

- Один и тот же Sequence как Primary Key Разных таблиц
 - Множество возможностей для полнофункциональной бизнес-логики
 - Гарантия непересекающихся ID для всех критичных бизнес-объектов
 - Журналирование всех таблиц в едином пространстве



- Преимущества единого инкремента для бизнес-логики
 - Глобальная ориентировочная шкала времени и размера всей системы
 - Возможность локализации элемента данных без названия таблицы
 - Анализ «скорости» роста инкремента как дополнительная опция мониторинга



Sequence как объединяющий несколько сущностей идентификатор

f_x

- Один Sequence как генератор Unique Разных таблиц (не в качестве PK)
 - Проектирование глобальной последовательности событий
 - Эффективный способ получения последовательных данных из разных таблиц
 - Журналирование всех связанных событий в едином пространстве



- Преимущества сквозного идентификатора для бизнес-логики
 - Удобный способ получения разрозненных данных в одном диапазоне
 - Оптимальная сортировка разрозненных данных через одно поле
 - Единый принцип организации данных из различных источников



Sequence как объединяющий несколько сущностей идентификатор

f_x

- Один Sequence как генератор Unique Разных таблиц (не в качестве PK)
 - Проектирование глобальной последовательности событий
 - Эффективный способ получения последовательных данных из разных таблиц
 - Журналирование всех связанных событий в едином пространстве



- Преимущества сквозного идентификатора для бизнес-логики
 - Удобный способ получения разрозненных данных в одном диапазоне
 - Оптимальная сортировка разрозненных данных через одно поле
 - Единый принцип организации данных из различных источников



Sequence как часть сгенерированного текстового идентификатора

- Составной идентификатор со смысловым текстом и численным инкрементом
 - Префикс ⇒ Тип объекта ⇒ Смысловое наполнение ⇒ Инкремент



- Текстовый компактный идентификатор с численным инкрементом
 - Префикс ⇔ Краткий случайный набор символов ⇔ Инкремент



Sequence как часть сгенерированного текстового идентификатора

- *inv_food_inc_1234*
- Обладает визуальным лаконизмом
- Предоставляет дополнительные возможности с данными из ИД
- Возможности секционирования по части из текста ИД
- Компактный размер по сравнению с большими текстами



Sequence как дробная часть временной уникальной последовательности

- Decimal идентификатор с информацией о последовательности и фактическом времени объекта
 - 1579874912.000001
 - UnixTime до секунды ⇔ Порядок (6) количества данных в секунду ⇔ Инкремент / 10e6



Sequence как дробная часть временной уникальной последовательности

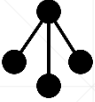
- *1579874912.001234*
- Обладает визуальным лаконизмом
- Компактный размер и тип данных для индексирования
- Высокая селективность данных
- Возможности секционирования по целой части числа
- Информационная «полезность» в виде даты и времени до секунды
- Включает преимущества сквозного РК/УК идентификатора



Сводное описание композитных Sequence идентификаторов

	Этап обработки	Описание	Области применения
<i>PK = 12345</i>	Единый Sequence как Primary Key	Сквозной PK идентификатор всех объектов всех сущностей системы	Единое пространство журналирования, оценка масштаба и роста системы
<i>UK = 12345</i>	Единый Sequence как Unique Key	Групповой UK для логически последовательных сущностей	Проектирование последовательных событий из разных источников данных
<i>inv_food_inc_1234</i>	Текстовый идентификатор с Sequence	Многоблочный текстовый набор именованных сокращений + Sequence	Функциональный компактный PK идентификатор с информативностью
<i>1579874912.000001</i>	Дробный временной с Sequence	Decimal, с UnixTime в целой части, и Sequence / значений в секунду в дробной	Сквозной численный идентификатор с последовательностью и датой

Преимущества композитных Sequence идентификаторов для реализации бизнес-логики



- Эффективный инструмент обеспечения уникальности и информативности
- Оптимизация использования памяти при работе с разными источниками
- Дополнительные возможности бизнес-логики с функциональным контентом
- Повышение производительности за счёт оптимальных типов данных
- Органичная часть кода для database-based бизнес-логики



