



Разработка серверной части приложений PostgreSQL Базовый курс

Преподаватель:
Валентин Степанов

День 2

- Приложение «Книжный магазин»
 - Схема данных и интерфейс
- SQL
 - Функции
 - Процедуры
 - Составные типы
- PL/pgSQL
 - Обзор и конструкции языка

Приложение «Книжный магазин»

Схема данных и интерфейс

Демонстрация



Книги

- ER-модель для высокоуровневого проектирования
 - сущности – понятия предметной области
 - связи – отношения между сущностями
 - атрибуты – свойства сущностей и связей



Схема данных

id	title	author	qty	operation
1	Муму	Тургенев Иван Сергеевич	10	+11
1	МУМУ	Тургенев Иван Сергеевич	10	-1
2	Отцы и дети	Тургенев Иван Сергеевич	4	+4
3	Трудно быть богом	Стругацкий Аркадий Натанович	7	+7
3	Трудно быть богом	Стругацкий Борис Натанович	7	0

10 = 11 - 1

7,0
или 0,7
или 7,7
?

- Данные дублируются
 - сложно поддерживать согласованность
 - сложно обновлять
 - сложно писать запросы

Схема данных (вариант)

entity	attribute	value
1	title	Муму
1	author	Тургенев Иван Сергеевич
1	qty	10
1	operation	+11
1	operation	-1
2	title	Отцы и дети
2	author	Тургенев Иван Сергеевич
2	qty	4
2	operation	+4
...

- Данные без схемы
 - поддержка согласованности на стороне приложения
 - сложно писать запросы
 - низкая производительность (множественные соединения)

Схема данных (вариант)

book_id	description
1	{ "title": "Муму", "authors": ["Тургенев Иван Сергеевич"], "qty": 10, "operations": [+11, -1] }
3	{ "title": "Трудно быть богом", "authors": ["Стругацкий Аркадий Натанович", "Стругацкий Борис Натанович"], "qty": 7, "operations": [+7] }
...	...

- Данные без схемы
 - поддержка согласованности на стороне приложения
 - сложно писать запросы (специальный язык)
 - индексная поддержка есть

Книги и операции

- Нормализация – уменьшение избыточности данных
 - разбиение крупных сущностей на более мелкие



Схема данных

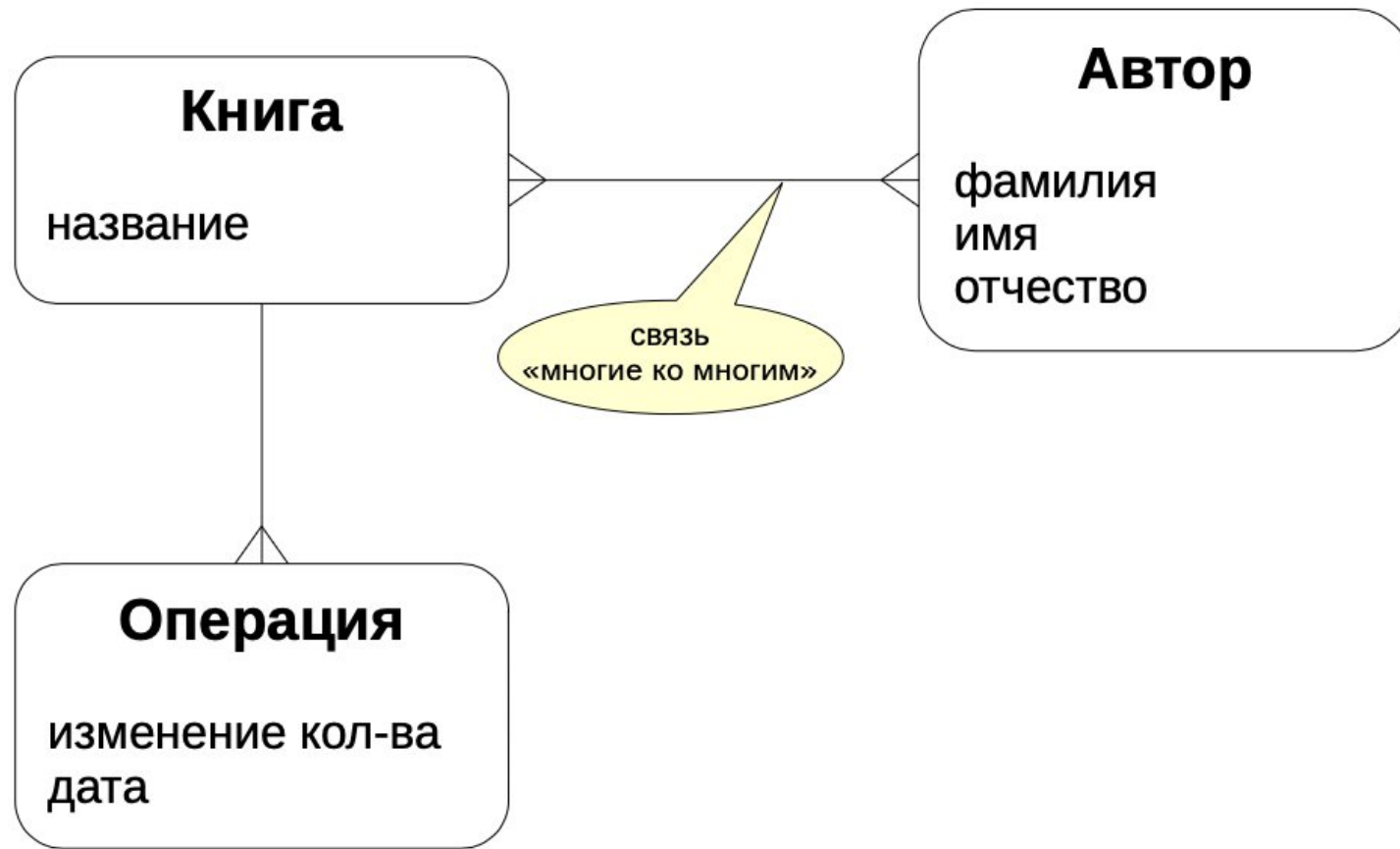
books

book_id	title	author
1	Муму	Тургенев Иван Сергеевич
2	Отцы и дети	Тургенев Иван Сергеевич
3	Трудно быть богом	Стругацкий Аркадий Натанович
3	Трудно быть богом	Стругацкий Борис Натанович

operations

operation_id	book_id	qty_change	date_created
1	1	+10	2020-07-13
2	1	-1	2020-08-25
3	3	+7	2020-07-13
4	2	+4	2020-07-13

Книги, авторы и операции



Организация интерфейса

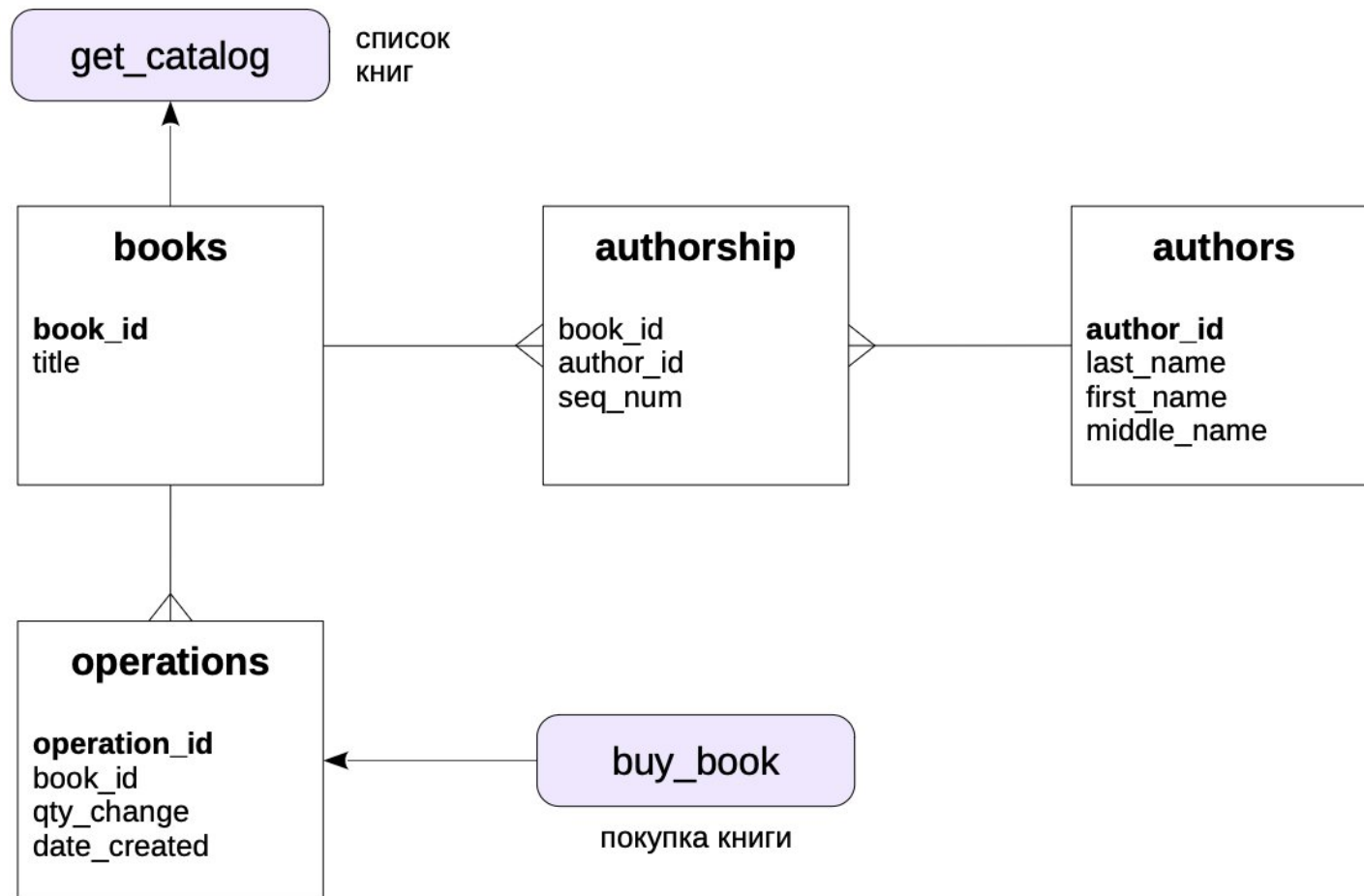
- Таблицы и триггеры

- чтение данных напрямую из таблицы (представления); запись данных напрямую в таблицу (представление), плюс триггеры для изменения связанных таблиц
- приложение должно быть в курсе модели данных, максимальная гибкость
- сложно поддерживать согласованность

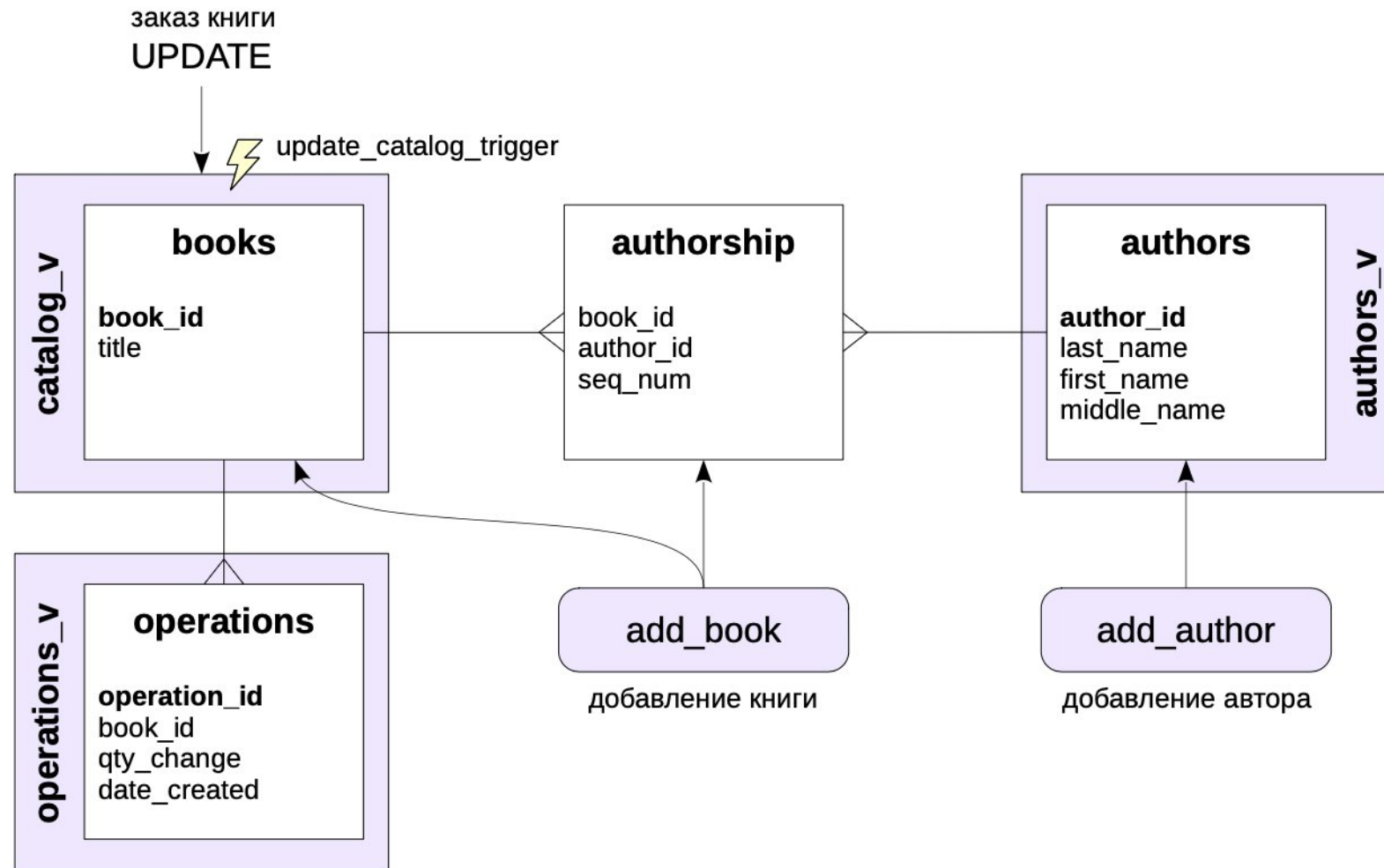
- Функции

- чтение данных из табличных функций; запись данных через вызов функций
- приложение отделено от модели данных и ограничено API
- большой объем работы по изготовлению функций-оберток, потенциальные проблемы с производительностью

Интерфейс магазина



Интерфейс «админки»



Итоги

- Проектирование баз данных – отдельная тема
 - теория важна, но не заменяет здравый смысл
- Отсутствие избыточности в данных делает работу удобнее и упрощает поддержку согласованности
- Для клиент-серверного интерфейса можно использовать таблицы, представления, функции, триггеры

Практика №6

1. В базе данных bookstore создайте схему bookstore. Настройте путь поиска к этой схеме на уровне подключения к БД.
 2. В схеме bookstore создайте таблицы books, authors, authorship и operations с необходимыми ограничениями целостности так, чтобы они соответствовали показанным в демонстрации.
 3. Вставьте в таблицы данные о нескольких книгах. Проверьте себя с помощью запросов.
 4. В схеме bookstore создайте представления authors_v, catalog_v и operations_v так, чтобы они соответствовали показанным в демонстрации.
- Проверьте, что приложение стало показывать данные на вкладках «Книги», «Авторы» и «Каталог».

Практика №7

1. Какие дополнительные атрибуты могут появиться у выделенных сущностей при развитии приложения?

2. Допустим, требуется хранить информацию об издательстве. Дополните ER-диаграмму и отобразите ее в таблицы.

3. Некоторые книги могут входить в серии (например, «Библиотека приключений»). Как изменится схема данных?

4. Пусть наш магазин стал торговать компьютерными комплектующими (материнскими платами, процессорами, памятью, жесткими дисками, мониторами и т. п.).

Какие сущности и какие атрибуты вы бы выделили? Учтите, что на рынке постоянно появляются новые типы оборудования со своими характеристиками.

SQL

Функции

Функции в базе данных

- Основной мотив: упрощение задачи
 - интерфейс (параметры) и реализация (тело функции)
 - о функции можно думать вне контекста всей задачи

	Традиционные языки	PostgreSQL
побочные эффекты	глобальные переменные	вся база данных (категории изменчивости)
модули	со своим интерфейсом и реализацией	пространства имен, клиент и сервер
сложности	накладные расходы на вызов (подстановка)	скрытие запроса от планировщика (подстановка, подзапросы, представления)

Общие сведения

- Объект базы данных
 - определение хранится в системном каталоге
- Основные составные части определения
 - имя
 - параметры
 - тип возвращаемого значения
 - тело
- Доступны несколько языков, в том числе SQL
 - код в виде строковой константы
 - интерпретируется при вызове
- Вызывается в контексте выражения

Вход и выход

- Входные значения
 - определяются параметрами с режимом IN и INOUT
- Выходное значение
 - определяется либо предложением RETURNS, либо параметрами с режимом INOUT и OUT
 - если одновременно указаны обе формы, они должны быть согласованы

Категории изменчивости

- Volatile
 - возвращаемое значение может произвольно меняться при одинаковых значениях входных параметров
 - используется по умолчанию
- Stable
 - значение не меняется в пределах одного оператора SQL
 - функция не может менять состояние базы данных
- Immutable
 - значение не меняется, функция детерминирована
 - функция не может менять состояние базы данных

Итоги

- Можно создавать собственные функции и использовать их так же, как и встроенные
- Функции можно писать на разных языках, в том числе SQL
- Изменчивость влияет на возможности оптимизации
- Иногда функция на SQL может быть подставлена в запрос

Практика №8

1. Создайте функцию `author_name` для формирования имени автора. Функция принимает три параметра (фамилия, имя, отчество) и возвращает строку с фамилией и инициалами.

Используйте эту функцию в представлении `authors_v`.

2. Создайте функцию `book_name` для формирования названия книги.

Функция принимает два параметра (идентификатор книги и заголовок) и возвращает строку, составленную из заголовка и списка авторов в порядке `seq_num`. Имя каждого автора формируется функцией `author_name`.

Используйте эту функцию в представлении `catalog_v`.

Проверьте изменения в приложении.

Практика №9

1. Напишите функцию, выдающую случайное время, равномерно распределенное в указанном отрезке.

Начало отрезка задается временной отметкой (timestamp), конец — либо временной отметкой, либо интервалом (interval).

2. В таблице хранятся номера автомобилей, введенные кое-как: встречаются как латинские, так и русские буквы в любом регистре; между буквами и цифрами могут быть пробелы. Считая, что формат номера «буква три-цифры две-буквы», напишите функцию, выдающую число уникальных номеров. Например, «K 123 XM» и «k123xm» считаются равными.

3. Напишите функцию, находящую корни квадратного уравнения.

SQL

Процедуры

Подпрограммы

- Функции
 - вызываются в контексте выражения
 - не могут управлять транзакциями
 - возвращают результат
- Процедуры
 - вызываются оператором CALL
 - могут управлять транзакциями
 - могут возвращать результат

Перегрузка

- Несколько подпрограмм с одним и тем же именем
 - подпрограммы различаются типами входных параметров
 - тип возвращаемого значения и выходные параметры игнорируются
 - подходящая подпрограмма выбирается во время выполнения в зависимости от фактических параметров
- Команда CREATE OR REPLACE
 - при несовпадении типов входных параметров создаст новую перегруженную подпрограмму
 - при совпадении — изменит существующую подпрограмму, но поменять тип возвращаемого значения нельзя

Полиморфизм

- Подпрограмма, принимающая параметры разных типов
 - формальные параметры используют полиморфные псевдотипы (например, `anyelement`)
 - конкретный тип данных выбирается во время выполнения по типу фактических параметров

Итоги

- Можно создавать и использовать собственные процедуры
- В отличие от функций, процедуры вызываются оператором CALL и могут управлять транзакциями
- Для процедур и функций поддерживаются перегрузка и полиморфизм

Практика №10

1. В таблице authors имена, фамилии и отчества авторов по смыслу должны быть уникальны, но это условие никак не проверяется. Напишите процедуру, удаляющую возможные дубликаты авторов.
2. Чтобы необходимость в подобной процедуре не возникала, создайте ограничение целостности, которое не позволит появляться дубликатам в будущем.

Практика №1 1

1. Получится ли создать в одной и той же схеме и имеющие одно и то же имя: 1) процедуру с одним входным параметром, 2) функцию с одним входным параметром того же типа, возвращающую некоторое значение? Проверьте.

2. В таблице хранятся вещественные числа (например, результаты каких-либо измерений). Напишите процедуру нормализации данных, которая умножает все числа на определенный коэффициент так, чтобы все значения попали в интервал от -1 до 1 .

Процедура должна возвращать выбранный коэффициент.

SQL

Составные типы

Составные типы

- Составной тип
 - набор именованных атрибутов (полей)
 - то же, что табличная строка, но без ограничений целостности
- Создание
 - явное объявление нового типа
 - неявно при создании таблицы
 - неопределенный составной тип record
- Использование
 - атрибуты как скалярные значения
 - операции со значениями составного типа: сравнение, проверка на NULL, использование с подзапросами

Параметры функций

- Функция может принимать параметры составного типа
- Способ реализации вычисляемых полей
 - взаимозаменяемость `table.column` и `column(table)`
- Другие способы
 - представления
 - столбцы `GENERATED ALWAYS`

«Однострочные» функции

- Возвращают значение составного типа
- Обычно вызываются в списке выборки запроса
- При вызове в предложении FROM возвращают однострочную таблицу

Табличные функции

- Объявляются как RETURNS SETOF или RETURNS TABLE
- Могут возвращать несколько строк
- Обычно вызываются в предложении FROM
- Можно использовать как представление с параметрами
 - особенно удобно в сочетании с подстановкой тела функции в запрос

Итоги

- Составной тип объединяет значения других типов
- Упрощает и обогащает работу функций с таблицами
- Позволяет создавать вычисляемые поля и представления с параметрами

Практика №12

1. Создайте функцию `onhand_qty` для подсчета имеющихся в наличии книг. Функция принимает параметр составного типа `books` и возвращает целое число.

Используйте эту функцию в представлении `catalog_v` в качестве «вычисляемого поля».

Проверьте, что приложение отображает количество книг.

2. Создайте табличную функцию `get_catalog` для поиска книг. Функция принимает значения полей формы поиска («имя автора», «название книги», «есть на складе») и возвращает подходящие книги в формате `catalog_v`.

Проверьте, что в «Магазине» начал работать поиск и просмотр.

Практика №13

1. Напишите функцию, переводящую строку, содержащую число в шестнадцатеричной системе, в обычное целое число.
2. Добавьте в функцию второй необязательный параметр — основание системы счисления (по умолчанию — 16).
3. Табличная функция `generate_series` не работает со строковыми типами. Предложите свою функцию для генерации последовательностей строк из заглавных английских букв.

PL/pgSQL

Обзор и конструкции языка

История PL/pgSQL

- Появился в версии 6.4 в 1998 году
 - устанавливается по умолчанию с версии 9.0
- Цели создания
 - простой язык для написания пользовательских функций и триггеров
 - добавить управляющие структуры к языку SQL
 - сохранить возможность использования любых пользовательских типов, функций и операторов
- Родословная: Oracle PL/SQL, Ада

Структура блока

- Метка блока
- Объявления переменных
 - область действия — блок
 - область видимости может перекрываться вложенными блоками, но можно сослаться на переменную, используя метку блока
 - любые типы SQL, ссылки на типы объектов (%TYPE)
- Операторы
 - управляющие конструкции
 - операторы SQL, кроме служебных
- Обработка исключительных ситуаций

Анонимные блоки

- Разовое выполнение процедурного кода
 - без создания хранимой подпрограммы
 - без возможности передать параметры
 - без возможности вернуть значение
- Оператор DO языка SQL

Подпрограммы PL/pgSQL

- Заголовок подпрограммы не зависит от языка
 - имя, входные и выходные параметры
 - для функций: возвращаемое значение, категория изменчивости
- Указание LANGUAGE plpgsql
- Возврат значений
 - оператор RETURN
 - присвоение значений выходным (INOUT, OUT) параметрам

Условные операторы

- IF
 - стандартный условный оператор
- CASE
 - похож на CASE языка SQL, но не возвращает значение
- Внимание: трехзначная логика
 - условие должно быть истинно; false и NULL не подходят

Циклы

- Цикл FOR по диапазону чисел
- Цикл WHILE с предусловием
- Бесконечный цикл
- Цикл может иметь метку, как блок
- Управление
 - выход из цикла (EXIT)
 - переход на новую итерацию (CONTINUE)

Вычисление выражений

- Любое выражение вычисляется в контексте SQL
 - выражение автоматически преобразуется в запрос
 - запрос подготавливается
 - переменные PL/pgSQL подставляются как параметры
- Особенности
 - можно использовать все возможности SQL, включая подзапросы
 - невысокая скорость выполнения, хотя разобранный запрос (и, возможно, план запроса) кешируются
 - неоднозначности при разрешении имен требуют внимания

Итоги

- PL/pgSQL – доступный по умолчанию, интегрированный с SQL, удобный и простой в использовании язык
- Управление подпрограммами на PL/pgSQL не отличается от работы с подпрограммами на других языках
- DO – команда SQL для выполнения анонимного блока
- Переменные PL/pgSQL могут использовать любые типы SQL
- Язык поддерживает обычные управляющие конструкции, такие как условные операторы и циклы

Практика №14

1. Измените функцию `book_name` так, чтобы длина возвращаемого значения не превышала 45 символов. Если название книги при этом обрезается, оно должно завершаться на троеточие.

Проверьте реализацию в SQL и в приложении; при необходимости добавьте книг с длинными названиями.

2. Снова измените функцию `book_name` так, чтобы избыточно длинное название уменьшалось на целое слово.

Проверьте реализацию.

Практика №15

1. Напишите PL/pgSQL-функцию, которая возвращает строку заданной длины из случайных символов.

2. Задача про игру в «наперстки».

В одном из трех наперстков спрятан выигрыш.

Игрок выбирает один из этих трех. Ведущий убирает один из двух оставшихся наперстков (обязательно пустой) и дает игроку возможность поменять решение, то есть выбрать второй из двух оставшихся.

Есть ли смысл игроку менять выбор или нет смысла менять первоначальный вариант?

Задание: используя PL/pgSQL, посчитайте вероятность выигрыша и для начального выбора, и для измененного.