

Оглавление

Глава 1. Введение	11
1.1. О книге	11
1.2. Учебная база данных	14
1.3. Благодарности автора	17
Глава 2. Общие табличные выражения	19
2.1. Запросы с несколькими общими табличными выражениями . . .	19
2.2. Рекурсивные общие табличные выражения	28
2.3. Массивы в общих табличных выражениях	49
2.4. Модификация данных в общем табличном выражении	69
2.5. Контрольные вопросы и задания	82
Глава 3. Аналитические возможности PostgreSQL	109
3.1. Агрегатные функции	109
3.2. Статистические функции	126
3.3. GROUPING SETS, CUBE и ROLLUP	137
3.4. Оконные функции	154
3.5. Гипотезирующие агрегатные функции	186
3.6. Контрольные вопросы и задания	187
Глава 4. Конструкция LATERAL команды SELECT	225
4.1. Подзапросы в предложении FROM	225
4.2. Вызовы функций в предложении FROM	234
4.3. Тип JSON и конструкция LATERAL	236
4.4. Контрольные вопросы и задания	244
Глава 5. Подпрограммы	263
5.1. Базовые сведения о функциях	263
5.2. Функции и зависимости между объектами базы данных	288
5.3. Функции, возвращающие множества строк	298
5.4. Функции с переменным числом аргументов	305
5.5. Конструкция LATERAL и функции	309
5.6. Категории изменчивости функций	314
5.7. Дополнительные сведения о функциях	347

Оглавление

5.8. Элементы теории принятия решений	357
5.9. Процедуры	367
5.10. Контрольные вопросы и задания	372
Предметный указатель	435

Содержание

Глава 1. Введение	11
1.1. О книге	11
1.2. Учебная база данных	14
1.3. Благодарности автора	17
Глава 2. Общие табличные выражения	19
2.1. Запросы с несколькими общими табличными выражениями	19
2.2. Рекурсивные общие табличные выражения	28
2.3. Массивы в общих табличных выражениях	49
2.3.1. Выявление циклов	49
2.3.2. Выявление множественных путей	60
2.3.3. Поиск маршрута между двумя городами	64
2.4. Модификация данных в общем табличном выражении	69
2.5. Контрольные вопросы и задания	82
2.5.1. Материализация общего табличного выражения может влиять на скорость выполнения всего запроса	82
2.5.2. А если главный запрос использует не все строки, порождаемые конструкцией WITH?	86
2.5.3. Изменяемая функция gandom: неожиданные эффекты	87
2.5.4. Не является ли промежуточная таблица лишним звеном?	89
2.5.5. Имя таблицы одно и то же, а совпадают ли наборы строк?	89
2.5.6. Почему рекурсивный запрос может зациклиться и что с этим делать?	90
2.5.7. Разница между UNION и UNION ALL в рекурсивном запросе	91
2.5.8. Порядком обхода иерархии управлять нельзя, а порядком вывода — можно	94
2.5.9. Для поиска циклов в графе есть встроенный синтаксис	95
2.5.10. В распавшейся на части иерархии тоже могут быть дефекты	95
2.5.11. Можно ли ускорить поиск маршрута между двумя городами?	98
2.5.12. Пути к вершинам-листьям можно найти, не создавая их список заранее	99
2.5.13. В массив можно «собрать» не только путь, но и стоимости ребер вдоль этого пути	100
2.5.14. Повторим все эксперименты на «большой» двоичной иерархии	101

15. Как выбрать альтернативные пути, имеющиеся в иерархии?	102
16. С помощью запросов, разработанных для иерархий, можно исследовать и граф общего вида	104
17. Главный запрос и все общие табличные выражения в WITH работают с одним и тем же снимком данных	104
18. Процедуру архивирования таблиц можно ускорить	106
19. В тексте главы была первая часть процедуры архивирования таблиц, а это — ее завершение	107

Глава 3. Аналитические возможности PostgreSQL 109

3.1. Агрегатные функции	109
3.1.1. Краткий обзор	109
3.1.2. Агрегирование в параллельном режиме	114
3.1.3. Агрегирование данных типа JSON	121
3.2. Статистические функции	126
3.3. GROUPING SETS, CUBE и ROLLUP	137
3.3.1. Группировка с помощью GROUPING SETS	137
3.3.2. Группировка с помощью ROLLUP	142
3.3.3. Группировка с помощью CUBE	145
3.4. Оконные функции	154
3.4.1. Использование агрегатных функций в качестве оконных	156
3.4.2. Способы формирования оконного кадра	164
3.4.3. Совместное использование оконных и агрегатных функций	173
3.4.4. Оконные функции общего назначения	176
3.5. Гипотезирующие агрегатные функции	186
3.6. Контрольные вопросы и задания	187
1. Битовые строки тоже можно агрегировать: функции bit_and и bit_or	187
2. Битовые строки тоже можно агрегировать: функция bit_xor	190
3. В каких частях запроса можно использовать агрегатные выражения?	190
4. Агрегирование в параллельном режиме	191
5. Агрегирование числовых данных, содержащихся в JSON-объектах	192
6. Как первичный ключ влияет на выбор группируемых столбцов?	193
7. Эксперимент с процентилями	195
8. Аргумент-массив функции можно сгенерировать, но есть нюансы	196
9. Небольшое статистическое исследование	197
10. Сопоставление конструкции UNION с конструкцией GROUPING SETS	199

11. Влияет ли порядок следования групп столбцов в конструкции GROUPING SETS на работу запроса?	200
12. Конструкция ROLLUP и ее отражение в плане запроса	201
13. Конструкция CUBE, предложение HAVING и функция GROUPING	203
14. Сопоставление конструкции UNION с конструкцией CUBE	206
15. Комбинирование конструкций GROUPING SETS, CUBE и ROLLUP	208
16. Оконная функция в предложении ORDER BY: можно ли обойтись без нее?	211
17. В запросе может быть несколько разных определений окна	213
18. Использование условия в определении раздела	215
19. Эксперименты с определениями окон и оконными функциями	217
20. Функции lead и lag	221
21. Оконные функции и родственные строки	223
22. Сравнение режимов формирования оконного кадра RANGE, ROWS и GROUPS	223
Глава 4. Конструкция LATERAL команды SELECT	225
4.1. Подзапросы в предложении FROM	225
4.2. Вызовы функций в предложении FROM	234
4.3. Тип JSON и конструкция LATERAL	236
4.4. Контрольные вопросы и задания	244
1. Оконная функция вместо конструкции LATERAL при отборе пассажиров для поощрения	244
2. Еще один вариант решения задачи отбора пассажиров для поощрения	245
3. Наглядное представление планировок салонов	247
4. Напоминает ли конструкция LATERAL коррелированный подзапрос?	250
5. Выявление пропусков в нумерации	250
6. Конструкция LATERAL вместо оконных функций	253
7. Использование конструкций LATERAL и WITH для многоступенчатых вычислений	255
8. Использование LATERAL во вложенных подзапросах	259
9. Функция JSON_TABLE	260
Глава 5. Подпрограммы	263
5.1. Базовые сведения о функциях	263
5.1.1. Создание функций	264
5.1.2. Перегрузка функций	275
5.1.3. Удаление функций	279

5.1.4. Функции, включающие несколько SQL-команд	280
5.1.5. Функции в стиле стандарта SQL	283
5.1.6. Значения NULL в качестве аргументов функции	285
5.2. Функции и зависимости между объектами базы данных	288
5.2.1. Зависимость объектов базы данных от функций	291
5.2.2. Зависимость функций от объектов базы данных	296
5.3. Функции, возвращающие множества строк	298
5.4. Функции с переменным числом аргументов	305
5.5. Конструкция LATERAL и функции	309
5.6. Категории изменчивости функций	314
5.6.1. Влияние изменчивости на выбор оптимального плана . . .	316
5.6.2. Видимость изменений	338
5.7. Дополнительные сведения о функциях	347
5.7.1. Подстановка кода функций в запрос	347
5.7.2. Функции и параллельный режим выполнения запросов . .	351
5.8. Элементы теории принятия решений	357
5.9. Процедуры	367
5.10. Контрольные вопросы и задания	372
1. Совместное использование параметров с модификатором OUT и предложения RETURNS	372
2. Значение параметра функции в качестве идентификатора в ее коде — это возможно?	372
3. Если имена параметров функции совпадают с именами столбцов таблицы	372
4. Символ одинарной кавычки в теле функции	373
5. Перегруженные функции и значения параметров по умолчанию	374
6. Аргументом функции может быть и значение NULL	377
7. Взаимосвязи объектов в базе данных	377
8. Могут ли параметры с модификаторами INOUT и OUT идти вперемежку?	378
9. Параметр VARIADIC идет последним. Почему?	379
10. Псевдонимы таблиц и столбцов: есть некоторая свобода	380
11. Когда в предложении FROM несколько табличных функций	381
12. Табличные функции в списке SELECT? Иногда можно, но все же лучше в предложении FROM	388
13. Стабильные функции могут зависеть от настроек сервера	396
14. Функция random. Как получить одни и те же значения при многократном вызове?	397

15. Функция random. Неожиданные результаты	398
16. Поиск корней квадратного уравнения	399
17. Параметр конфигурации сервера можно изменить на время выполнения функции	402
18. Видит ли изменчивая функция изменения, произведенные конкурентной транзакцией?	403
19. А если вызвать изменчивую функцию из стабильной или постоянной? .	405
20. Не только багаж, но и питание	406
21. Подведение итогов по операции бронирования: подстановка кода функции в запрос	406
22. Подстановка в запрос кода скалярной функции	408
23. Пользовательские функции в индексных выражениях	411
24. Иллюстрация использования системного каталога pg_depend	414
25. Подстановка кода функций в запрос: детальные эксперименты	417
26. Как отобрать из Парето-оптимального множества единственную альтернативу?	418
27. Функция pareto: результат попарного сравнения альтернатив в другой форме	419
28. Составные значения в качестве аргументов функций	419
29. ROWS — характеристика количества строк, возвращаемых функцией . .	421
30. COST — характеристика стоимости вычисления функции	426
31. Вычисляемые столбцы, сохраняемые в таблице	427
32. Сортировка массивов разной размерности	430
33. Формирование русского или английского алфавита	432
34. Функция для определения степени заполнения самолетов	433

Глава 1

Введение

1.1. О книге

Язык SQL является общепринятым средством взаимодействия с данными. Недавно этому языку исполнилось полвека, но реляционные технологии не только не собираются сдавать своих позиций, но и продолжают развиваться и распространяться все шире. Знание SQL необходимо всем, кто имеет отношение к базам данных: и разработчикам приложений, и аналитикам, и администраторам. Его поддерживают все реляционные СУБД, и PostgreSQL не является исключением. В этой системе весьма полно реализованы требования стандарта к языку SQL и представлен целый ряд «фирменных» расширений.

Несколько лет назад при поддержке компании Postgres Professional была издана книга «PostgreSQL. Основы языка SQL»¹. В ней рассматривались основы языка и такие вопросы, как транзакции и повышение производительности. Однако современный язык SQL выходит далеко за эти рамки, предлагая множество различных команд и возможностей, зачастую сложных для освоения, но открывающих путь к эффективной работе с данными. Поэтому и возникла идея написать продолжение базового учебника.

Вот такие темы вошли в эту книгу.

Глава 2 посвящена общим табличным выражениям. Они кратко рассматривались в первой книге. Общие табличные выражения — очень полезное средство, которое позволяет, например, в одном запросе обрабатывать иерархические данные и даже графы общего вида. В качестве иллюстраций рассмотрены структура узлов и деталей самолета, штатное расписание компании, сеть маршрутов между городами.

¹ Моргунов, Е. П. PostgreSQL. Основы языка SQL. — СПб. : БХВ-Петербург, 2018. — 336 с. — ISBN 978-5-9775-4022-3.

В **главе 3** представлены аналитические возможности PostgreSQL. Рассмотрение начинается с агрегатных функций. Наиболее употребительные из них были показаны в базовом учебнике. Продемонстрировано использование основных статистических функций для вычисления дисперсии, среднеквадратического отклонения, моды, медианы. Важное место в этой главе занимают группировки данных. Конструкции GROUPING SETS, CUBE и ROLLUP позволяют свести несколько запросов с обычными группировками к одному. Большое внимание уделено оконным функциям, в частности особенностям формирования оконного кадра в разных режимах. Завершается глава обращением к гипотезирующим агрегатным функциям, то есть функциям, работающим с гипотетическими множествами строк.

Небольшая **глава 4** демонстрирует конструкцию LATERAL команды SELECT. В отличие от обычного соединения в предложении FROM, она позволяет подзапросу обращаться к полям из текущей строки другой таблицы, фактически образуя цикл в рамках одного запроса. Показаны также вызовы стандартных функций в предложении FROM.

Глава 5 посвящена программированию на стороне сервера. Рассмотрение начинается с простых функций, возвращающих одну строку, в том числе перегруженных. Затем вводятся табличные функции, а также функции с переменным числом параметров. Показано применение модификаторов IN, OUT и INOUT. Эта глава — самая большая в книге, поскольку написание и применение функций имеет множество нюансов. Функции могут зависеть от других объектов базы данных, категория изменчивости влияет на решения планировщика при оптимизации запросов и на видимость изменений, при выполнении ряда условий код функции может быть подставлен в запрос, а еще функции могут участвовать в параллельном выполнении. В качестве иллюстрации применения функций для поддержки принятия решений показан выбор наилучших альтернатив с помощью формирования множества Парето. Основной материал главы посвящен функциям, но в завершающем разделе обсуждаются и процедуры, которые во многом схожи с функциями, но имеют целый ряд отличий.

Книгу можно использовать как в процессе изучения технологий баз данных в образовательном учреждении (в бакалавриате и магистратуре), так и для самостоятельной подготовки.

Большое внимание в учебнике уделяется заданиям и упражнениям. Они различаются по объему и сложности, а также имеют разную специфику: многие предполагают написание SQL-запроса, однако есть и такие, которые заключаются в проведении экспериментов и изучении взаимосвязей между объектами базы данных. Упражнения зачастую дополняют и развивают основной текст главы, в них обсуждаются различные тонкости, частные случаи, полезные приемы и поучительные ситуации. Упражнения в конце каждой главы расположены в основном в соответствии с порядком изложения материала. Но есть и комплексные упражнения, выполнение которых требует освоения нескольких разделов или даже всей главы в целом.

Многие положения книги подкрепляются ссылками на соответствующие разделы документации, ведь именно она — первоисточник знаний. Эти разделы стоит просматривать, поскольку книга не является заменой документации и не пытается ее пересказывать.

Для учебных заведений можно предложить следующее распределение времени, исходя из того, что на изучение материала пособия будет отведено 36 академических часов:

Глава 2. Общие табличные выражения	6 часов
Глава 3. Аналитические возможности PostgreSQL	10 часов
Глава 4. Конструкция LATERAL команды SELECT	4 часа
Глава 5. Подпрограммы	16 часов

Для изучения материала учебника необходимо установить СУБД PostgreSQL (или Postgres Pro Standard) и развернуть учебную базу данных «Авиаперевозки». Скачать PostgreSQL можно с официального сайта [postgresql.org/download](https://www.postgresql.org/download), на котором приведена и инструкция по установке. Для использования Postgres Pro Standard в учебных заведениях нужно получить академическую лицензию, которую компания Postgres Professional предоставляет бесплатно¹. Учебная база данных и ее полное описание доступны на сайте компании: postgrespro.ru/education/demodb. Необходима версия от 15.08.2017, вариант small.

На странице книги (postgrespro.ru/education/books/advancedsql) предлагается виртуальная машина, в которой уже установлены СУБД и учебная база данных.

¹ Для этого требуется отправить заявку на academy@postgrespro.ru.

Все SQL-запросы, приведенные в книге, были выполнены в среде PostgreSQL версии 17. Ссылки на разделы документации относятся именно к этой версии.

Большинство запросов, скорее всего, будут корректно выполняться и в более новых версиях PostgreSQL. В старых версиях результаты могут отличаться от приведенных в книге. Например, в книге используется появившийся в версии 17 перегруженный вариант функции `random`, позволяющий задать диапазон случайных значений.

Ряд запросов, представленных в книге, содержат столбцы типа `timestampz`. Такие запросы могут возвращать результаты, отличающиеся от приведенных в книге для часового пояса `Asia/Krasnoyarsk (+07)`, поскольку зависят от настройки часового пояса на компьютере читателя.

Команды, вводимые пользователем как в среде операционной системы, так и в среде утилиты `psql`, выделяются в пособии полужирным моноширинным шрифтом, а результаты их выполнения — светлым. Например:

```
postgres$ psql -d demo -U postgres
```

или

```
SELECT min( days_of_week ), max( days_of_week )
FROM routes;
 min | max
-----+-----
 {1} | {7}
(1 строка)
```

1.2. Учебная база данных

В качестве учебной базы данных используется база данных «Авиаперевозки», разработанная в компании Postgres Professional. Она представлена в трех вариантах: малом (данные о полетах за месяц), среднем (полеты за квартал) и большом (за год). Эта база данных имеет целый ряд привлекательных качеств:

- данные в ней правдоподобны (например, частоты фамилий и имен пассажиров аналогичны тем, что имеют место в России);

- данных много (даже в малом варианте базы данных есть таблицы, содержащие сотни тысяч записей, а в одной из них — больше миллиона);
- база не слишком простая и не слишком сложная (число сущностей не превышает десятка);
- проиллюстрированы разные приемы проектирования (связь «один к одному», суррогатный ключ, контролируемая избыточность и др.);
- используются разнообразные типы данных (числа, строки, даты и время, интервалы, массивы, JSON).

Легенда такова. Некая авиакомпания совершает пассажирские авиаперевозки по России. Все модели авиапарка компании собраны в представлении «Самолеты» (aircrafts), созданном на основе таблицы aircrafts_data. Коды и названия моделей соответствуют реальной практике. В таблице «Места» (seats) записаны схемы салонов, одинаковые для всех самолетов одной модели.

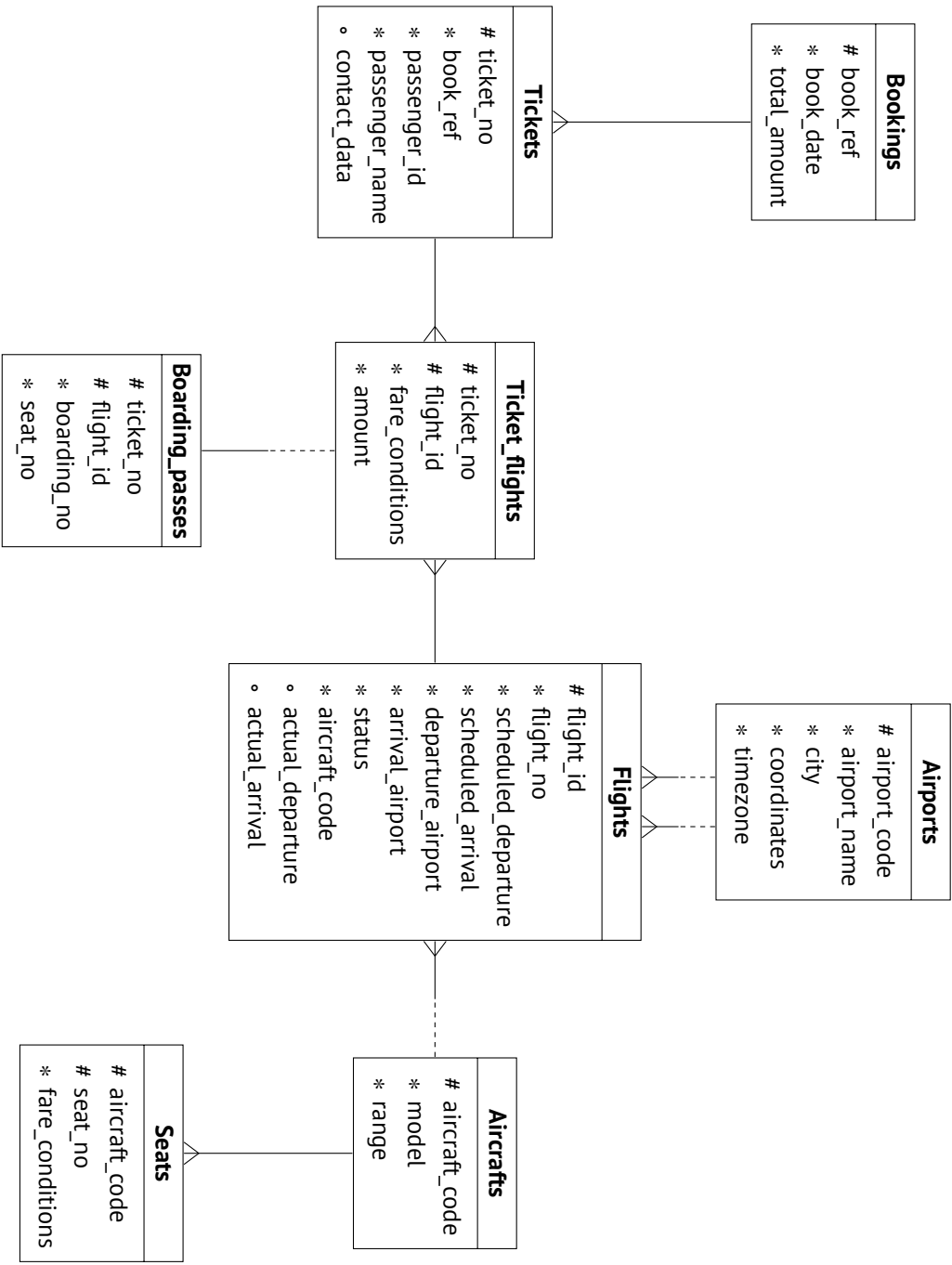
Представление «Аэропорты» (airports) на основе таблицы airports_data показывает область присутствия авиакомпании. Используются коды аэропортов, присваиваемые международной ассоциацией ИАТА. Расписание рейсов содержится в таблице «Рейсы» (flights). Ее первичным ключом является столбец «Идентификатор рейса» (flight_id). Это суррогатный ключ. Представление «Рейсы» (flights_v), созданное на основе этой и ряда других таблиц, выдает гораздо больше информации, например показывает местное время вылета и прибытия рейсов в соответствующих аэропортах.

Маршруты, то есть пары аэропортов, между которыми авиакомпания выполняет рейсы, можно увидеть в представлении «Маршруты» (routes).

Пассажиры бронируют билеты. Эти операции отражаются в таблице «Бронирования» (bookings). Столбец «Номер бронирования» (book_ref) содержит коды операций, состоящие из шести шестнадцатеричных цифр.

В каждой операции бронирования могут быть оформлены один или несколько билетов (возможно, на разных пассажиров), которые записываются в таблицу «Билеты» (tickets). Номера билетов представляют собой тринадцатизначные строки, состоящие из цифр.

Перелеты, входящие в оформленные билеты, записываются в таблицу «Перелеты» (ticket_flights). Сумма значений столбца «Стоимость перелета» (amount)



в записях, соответствующих конкретной операции бронирования, должна быть равна значению поля «Полная сумма бронирования» (`total_amount`) в строке таблицы «Бронирования» (`bookings`), описывающей эту операцию. Это пример контролируемой избыточности.

Регистрация на рейсы отражается в таблице «Посадочные талоны» (`boarding_passes`). Она связана с таблицей «Перелеты» (`ticket_flights`) отношением «один к одному». Столбец «Статус рейса» принимает только ограниченный набор значений, например `Departed` или `Arrived`.

Все перечисленные таблицы и представления созданы в схеме `bookings`. В ней же имеется функция `now`, представляющая момент выгрузки данных, или *текущий момент*, который можно использовать в запросах вместо стандартной функции `now`.

1.3. Благодарности автора

Хотя на обложке книги указан один человек, такая работа не выполняется единолично. Многие сотрудники компании Postgres Professional помогали автору на разных этапах ее выполнения.

Огромный вклад в успешное завершение этого труда внес Егор Рогов. Он проделал большую работу в качестве редактора: предложил целый ряд структурных изменений текста книги, множество более точных формулировок ключевых понятий, улучшений SQL-запросов и модификаций упражнений, приведенных в книге. Егор выполнил верстку текста, подготовил предметный указатель, нарисовал иллюстрации, помогающие разобраться в сложных понятиях и конструкциях языка SQL.

Павел Толмачев и Павел Лузанов просматривали черновые варианты текста книги и давали полезные комментарии. Включить в книгу главу «Конструкция LATERAL команды SELECT» предложил Павел Лузанов.

В обсуждении названия книги принял участие и высказал полезные идеи Александр Мелешко.

Содержание книги оттачивалось в ходе проведения пробного курса по ее материалу со студентами Алтайского государственного технического университета

и Алтайского государственного университета. Полезные замечания высказывали Михаил Козлов, Олег Целебровский и Павел Гилев, ставшие сотрудниками компании.

И конечно, книга не состоялась бы без всесторонней поддержки и внимания со стороны руководителей компании Postgres Professional Олега Бартунова и Ивана Панченко.

Замечания и предложения можно направлять по адресу edu@postgrespro.ru.

Мы надеемся, что это учебное пособие внесет вклад в повышение уровня квалификации читателя и расширение его профессионального кругозора.

Конец ознакомительного фрагмента.

Приобрести книгу можно

в интернет-магазине

«Электронный универс»

e-Univers.ru