ГЛАВА 4

Работа с запросами

Краткое содержание

- ✓ Методы записи запросов
- ✓ Основы SQL-запросов
- ✓ Использование псевдонима
- ✓ Сортировка данных по алфавиту
- ✓ Ограничение выборки данных
- ✓ Контрольные вопросы

SQL — это мощный и надежный язык, предлагающий пользователю широкий спектр команд. В действительности язык SQL содержит их гораздо больше, чем мы показываем в этой книге. Самые простые и удобные команды вы сможете сразу же начать изучать на практике, а не только в теории. В следующей главе вы познакомитесь с основами создания хорошего запроса и форматирования результатов. Изучив эту главу, вы научитесь выбирать отдельные поля из конкретной базы данных и отображать их в алфавитном порядке. Давайте начнем!

Добавление комментариев к запросам

Прежде чем составить свой первый SQL-запрос, рассмотрим, как писать комментарии. Комментарии — это простые предложения, которые помогают понять логику вашего SQL-запроса. Использование комментариев в SQL очень полезно, так как они поясняют сложную логику запроса.

В SQL комментарии бывают двух типов. Однострочные начинаются с двух дефисов (--). Пример ниже демонстрирует комментарий, созданный в строке 1 (рис. 32).

Многострочные комментарии начинаются с сочетания символов /* и заканчиваются символами */. Все, что находится между открывающим и закрывающим символами, — это комментарии (рис. 33).

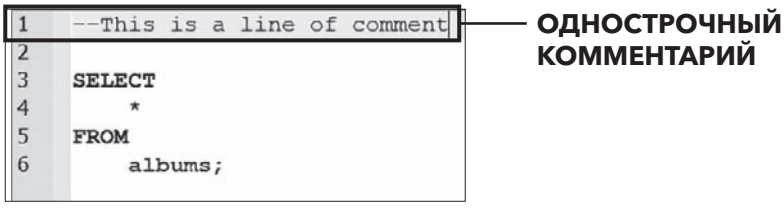

Рис. 32

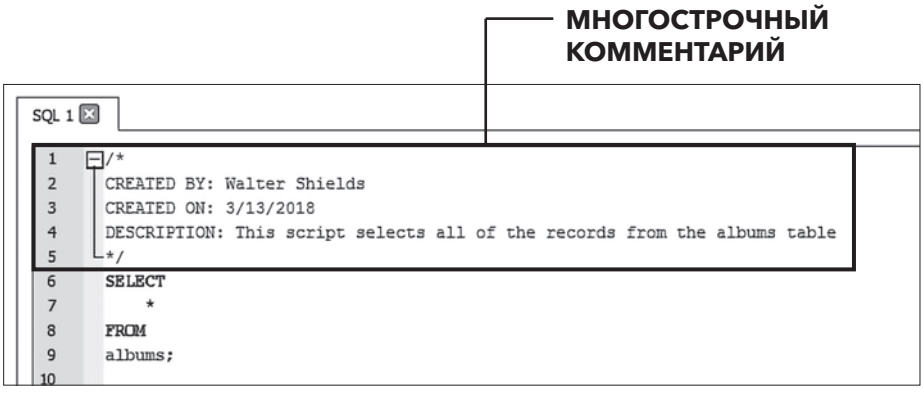

Рис. 33

В примере приведена стандартная информация, которую полезно включать в блок комментариев. Разработчик, дата разработки и описание имеют большое значение для всех, кто сталкивается с SQL-запросом или сценарием.

НА МОЙ ВЗГЛЯД,

разработчики частенько игнорируют комментарии в SQL. В следующих главах мы не используем комментарии только для краткости. При работе с реальными базами данных использование комментариев экономит время, которое пришлось бы потратить на написание дополнительных запросов, чтобы прояснить работу базы данных. Комментарии особенно важны, когда ваши запросы используются другими разработчиками.

Общая структура запроса

Написать запрос — это то же самое, что задать вопрос на любом человеческом языке. Большое значение имеют формулировка, детали и порядок слов. Чем детальнее наш вопрос, тем точнее будет ответ.

При создании SQL-запроса необходимо учитывать следующие пять моментов.

1. С какой базой данных мы работаем?
2. Из какой таблицы в этой базе данных нам необходимо извлечь данные?
3. Какие поля в этой таблице нас интересуют?
4. Хотим ли мы исключить какие-либо данные, отфильтровать или исключить какой-либо диапазон или период времени?
5. Как сформулировать наш запрос одним простым предложением на человеческом языке?

Цель этих вопросов — построить взаимосвязь между человеческим языком, на котором мы обычно говорим или пишем, и языком SQL. Если вы работаете аналитиком данных, то обычные вопросы о бизнесе, которые вам задают, необходимо преобразовать в операторы SQL. После получения результатов запроса следует преобразовать их обратно в доступный для всех человеческий язык. Таков принцип работы.

ПРИМЕЧАНИЕ

Если у вас возникли проблемы с запросом, ответьте на пять вопросов, приведенных выше. И только потом пишите свой запрос.

Пишем свой первый запрос

Чтобы написать свой первый запрос, используйте существующую вкладку Execute SQL (Выполнить SQL-запрос), обозначенную SQL 1, или откройте новую вкладку панели запросов, как если бы вы открывали новую вкладку веб-браузера. Щелкните на значке Open Tab (Открыть вкладку) (рис. 34).

Итак, новая панель запросов открыта! В первую очередь пишем блок комментариев:

```
/*
CREATED BY: <ваше имя>
CREATED ON: <дата>
DESCRIPTION: <краткое описание запроса, например вопрос № 5>
*/
```

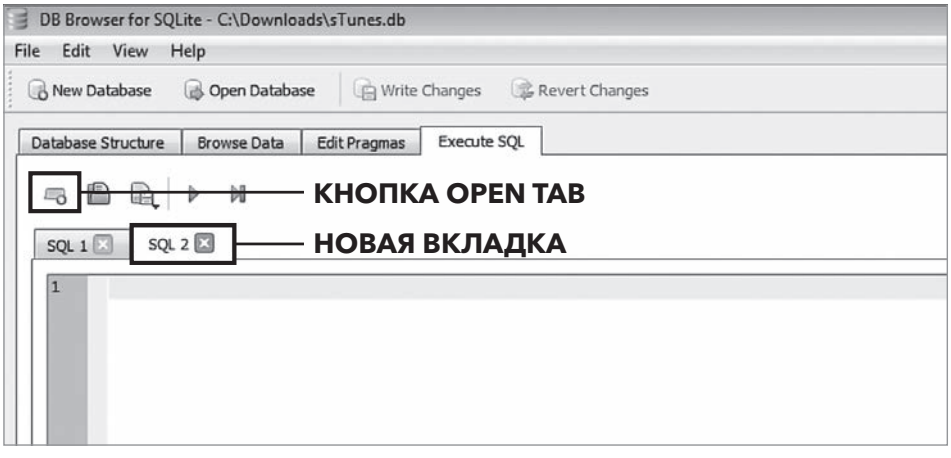

Рис. 34

Закончив блок комментариев, займемся написанием запроса SQL. Для запроса нам необходим понятный вопрос. В главе 2 мы рассмотрели пример рабочего сценария для нашей книги: вы аналитик данных компании sTunes. Как вы уже знаете из главы 3, компания sTunes специализируется на онлайн-продажах музыкальной продукции и имеет цифровую библиотеку исполнителей, треков и альбомов, а также список клиентов, которые приобрели музыкальные произведения. Предположим, служба поддержки sTunes желает разослать новую рекламу всем своим клиентам. В этом случае службе поддержки клиентов необходимо знать, обновлен ли список клиентских контактов, поэтому они наверняка поинтересуются, можем ли мы дать им полный список имен, фамилий и адресов электронной почты клиентов (если имеются) из базы данных. Как мы ответим на этот вопрос? Вернемся к пяти базовым моментам, которые следует учитывать при составлении запроса и о которых речь шла ранее в этой главе.

1. **С какой базой данных мы работаем?**

   В этом случае мы будем работать только с одной базой данных. База данных sTunes уже должна быть открыта в DB Browser. Если вы после изучения главы 3 закрыли браузер, откройте снова его, а также файл базы данных sTunes.

2. **Из какой таблицы в этой базе данных нам необходимо извлечь данные?**

   Нам необходимо получить информацию о клиентах. Просматривая вкладку Database Structure (Структура базы данных), мы видим, что у нас есть таблица с именем `customers (клиенты)`. Это то, что нам нужно!

3. **Какие поля в этой таблице нас интересуют?**

   Ответить на этот вопрос можно, заглянув во вкладку Browse Data (Просмотр данных). Щелкните на этой вкладке и в раскрывающемся меню выберите таблицу `customers` — таблица содержит поля для имени, фамилии и электронной почты.

4. **Хотим ли мы исключить какие-либо данные, отфильтровать или исключить какой-либо диапазон или период времени?**

   В данном случае служба поддержки sTunes желает получить список всех клиентов, поэтому лучше ничего не исключать.

5. **Как сформулировать наш запрос одним простым предложением на человеческом языке?**

   С помощью запроса нам необходимо осуществить выборку следующей информации из таблицы `customers`: имя, фамилия и адрес электронной почты.

Сначала добавьте блок комментариев, затем добавьте условие `FROM customers`. Оно определяет, в какой таблице искать данные. Далее перед условием `FROM` введите оператор `SELECT` и необходимые имена полей из таблицы `customers`. Имя каждого поля отделяется запятой. Полученный код будет выглядеть следующим образом:

```
/*
CREATED BY: Уолтер Шилдс
CREATED ON: 03/13/2018
DESCRIPTION: Данный запрос осуществляет выборку полей имени, фамилии,
электронной почты из таблицы customers (клиентов).
*/

SELECT
   FirstName,
   LastName,
   Email
FROM
   customers;
```

Запустите запрос на выполнение, щелкнув на кнопке воспроизведения Execute SQL (Выполнить SQL-запрос), расположенной на панели меню. Результаты запроса отобразятся ниже на панели результатов (рис. 35). Панель сообщений также показывает, что в результате выполнения запроса вернулось 59 строк (или записей) за 3 миллисекунды.

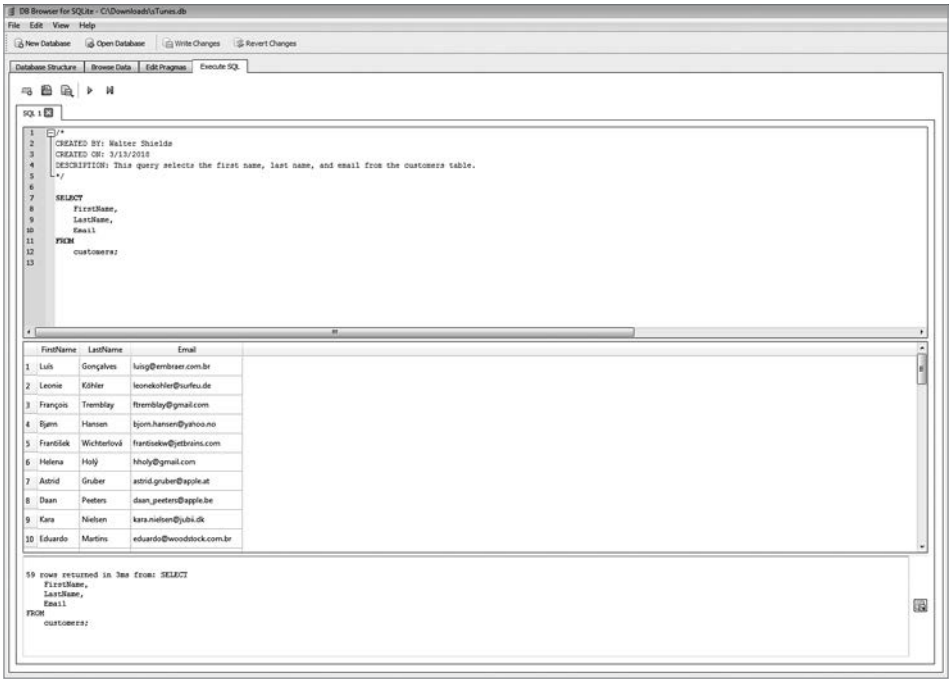

Рис. 35

Практические задания

✶ Добавьте в запрос еще одно поле из таблицы `customers`. В список рассылки добавьте поле `Company` или `Phone`. Не забудьте запятую!

Синтаксис и соглашение о кодировании

Как уже упоминалось в главе 1, все запросы должны соответствовать определенному синтаксису, чтобы их мог понять браузер SQL. Однако при написании запросов необходимо учитывать еще кое-что. Важно, чтобы другие пользователи базы данных могли легко понять ваши запросы. Набор принципов, которые задают стиль, методы написания запросов и т. д., известен как *соглашение о кодировании*. Для разных сред баз данных соглашения о кодировании различаются. В этом разделе мы приводим соглашения о кодировании, используемые в этой книге.

В предыдущих примерах после ключевого слова SELECT мы использовали символ * вместо указания отдельных полей. С помощью специального символа * осуществляется выборка всех записей из таблицы. В некоторых случаях этот символ полезен. Однако чаще рекомендуется указывать необходимые поля.

Здесь точку с запятой в конце оператора ставить необязательно, так как мы пишем только один оператор SQL. Точка с запятой обозначает конец оператора SQL. Поскольку большинство SQL-запросов, используемых в этой книге, представляют собой отдельные операторы, точку с запятой мы будем в дальнейшем опускать.

В нашем примере в условии SELECT мы определили три поля для отображения. Каждое поле необходимо отделять запятой (кроме последнего). Отсутствие запятой между полями или наличие ее после последнего поля — это распространенные синтаксические ошибки, о которых вы получите сообщение на панели результатов.

Обратите внимание, что код состоит из нескольких строк. Также запрос можно написать в одной строке, и браузер SQL по-прежнему распознает код и вернет результаты. Однако запросы рекомендуется разделять на условия, при этом каждое условие необходимо писать с новой строки. В следующих главах наши запросы станут намного длиннее и будут содержать несколько условий. Использование отступов и пробелов в запросах повышает удобочитаемость и значительно упрощает понимание сложной логики запроса.

НАПОМИНАНИЕ

Условие – это часть инструкции SQL, которая начинается со специального ключевого слова (SELECT, FROM и т. д.) и может включать дополнительные параметры и операторы.

Использование псевдонима

Как правило, технический язык базы данных отличается от общепринятого языка. Часто вам придется работать со старыми базами данных или базами данных, имена полей которых давно не обновлялись. *Псевдонимы* SQL используются для присвоения таблице или столбцу в таблице временного имени. Псевдонимы часто используются для того, чтобы сделать имена столбцов более удобочитаемыми. Псевдоним существует только на время выполнения запроса.

В следующем примере мы рассмотрим различные способы создания псевдонима для выбранных имен полей из таблицы `customers`. Псевдоним в базе данных всегда указывают сразу после имени поля. Псевдонимы обычно связаны с ключевым словом `AS`, однако в большинстве реализаций РСУБД ключевое слово `AS` между именем поля и псевдонимом использовать необязательно.

```
/*
CREATED BY: Уолтер Шилдс
CREATED ON: 13.03.2018
DESCRIPTION: Данный запрос осуществляет выборку полей имени, фамилии,
электронной почты и номера телефона из таблицы customers (клиенты)
и демонстрирует четыре различных способа использования псевдонима.
*/

SELECT
   FirstName AS 'First Name',
   LastName AS [Last Name],
   Email AS EMAIL
   Phone CELL
FROM
   customers
```

В данном запросе для первых трех полей мы использовали ключевое слово `AS`. Затем для поля `Phone`, которое мы переименовали в `CELL`, ключевое слово `AS` мы опустили. Если созданный вами псевдоним содержит несколько слов (например, имя и фамилию), его необходимо разграничить, в данном случае либо одинарными кавычками `''`, либо квадратными скобками `[]`, как показано в примере. Поскольку псевдонимы `EMAIL` и `CELL` представляют собой отдельные слова, нет необходимости их заключать в кавычки или скобки.

ПРИМЕЧАНИЕ

В SQL существует множество вариантов синтаксиса псевдонимов. Другие РСУБД могут не распознавать все перечисленные здесь варианты псевдонимов. Если при выполнении запроса обнаружена синтаксическая ошибка, проверьте, как вы указали псевдоним.

Как показано на рис. 36, в расположенных слева выходных данных имена полей не меняются. Расположенные справа выходные данные показывают, как при использовании псевдонимов изменились имена столбцов. При добавлении псевдонима данные в базе данных не изменяются. Псевдонимы изменяют только способ отображения полей на панели результатов.

БЕЗ ПСЕВДОНИМА

	FirstName	LastName	Email	Phone
1	Luís	Gonçalves	luisg@embraer.com.br	+55 (12) 3923-5555
2	Leonie	Köhler	leonekohler@surfeu.de	+49 0711 2842222
3	François	Tremblay	ftremblay@gmail.com	+1 (514) 721-4711
4	Bjørn	Hansen	bjorn.hansen@yahoo.no	+47 22 44 22 22
5	František	Wichterlová	frantisekw@jetbrains.com	+420 2 4172 5555
6	Helena	Holý	hholy@gmail.com	+420 2 4177 0449
7	Astrid	Gruber	astrid.gruber@apple.at	+43 01 5134505
8	Daan	Peeters	daan_peeters@apple.be	+32 02 219 03 03
9	Kara	Nielsen	kara.nielsen@jubii.dk	+453 3331 9991
10	Eduardo	Martins	eduardo@woodstock.com.br	+55 (11) 3033-5446

С ПСЕВДОНИМОМ

	First Name	Last Name	EMAIL	CELL
1	Luís	Gonçalves	luisg@embraer.com.br	+55 (12) 3923-5555
2	Leonie	Köhler	leonekohler@surfeu.de	+49 0711 2842222
3	François	Tremblay	ftremblay@gmail.com	+1 (514) 721-4711
4	Bjørn	Hansen	bjorn.hansen@yahoo.no	+47 22 44 22 22
5	František	Wichterlová	frantisekw@jetbrains.com	+420 2 4172 5555
6	Helena	Holý	hholy@gmail.com	+420 2 4177 0449
7	Astrid	Gruber	astrid.gruber@apple.at	+43 01 5134505
8	Daan	Peeters	daan_peeters@apple.be	+32 02 219 03 03
9	Kara	Nielsen	kara.nielsen@jubii.dk	+453 3331 9991
10	Eduardo	Martins	eduardo@woodstock.com.br	+55 (11) 3033-5446

Рис. 36

Практические задания

- Добавьте к запросу еще одно поле и укажите для него псевдоним.
- Попрактикуйтесь в изменении синтаксиса псевдонима, попробуйте исключить ключевое слово `AS`. Проанализируйте результат.

ПРИМЕЧАНИЕ

Не используйте никакие ключевые слова SQL в качестве псевдонимов. Это вызовет путаницу или синтаксические ошибки. РСУБД может интерпретировать данный псевдоним как команду.

Условие ORDER BY

Предположим, что службе поддержки необходим список клиентов sTunes. Целесообразно упорядочить полученные результаты по фамилии клиентов. Чтобы сделать это, надо использовать новое условие после условия `FROM`. Условие `ORDER BY` используется для сортировки данных в порядке возрастания или убывания на основе одного или нескольких столбцов. По умолчанию данные будут отсортированы в порядке возрастания от A до Z. Специальное ключевое слово `ASC`, определяющее порядок сортировки, необязательно. Для сортировки в порядке убывания от Z до A необходимо после сортируемого поля добавить ключевое слово `DESC`. Например, запрос `ORDER BY LastName DESC` сортирует столбец с псевдонимом `LastName` в порядке убывания.

```
/*
CREATED BY: Уолтер Шилдс
CREATED ON: 13.03.2018
DESCRIPTION: Данный запрос осуществляет выборку полей имени, фамилии
и электронной почты из таблицы customers (клиенты), отсортированных
по фамилии (Last Name).
*/

SELECT
   FirstName AS [First Name],
   LastName AS [Last Name],
   Email AS [EMAIL]
FROM
   customers
ORDER BY
   LastName ASC
```

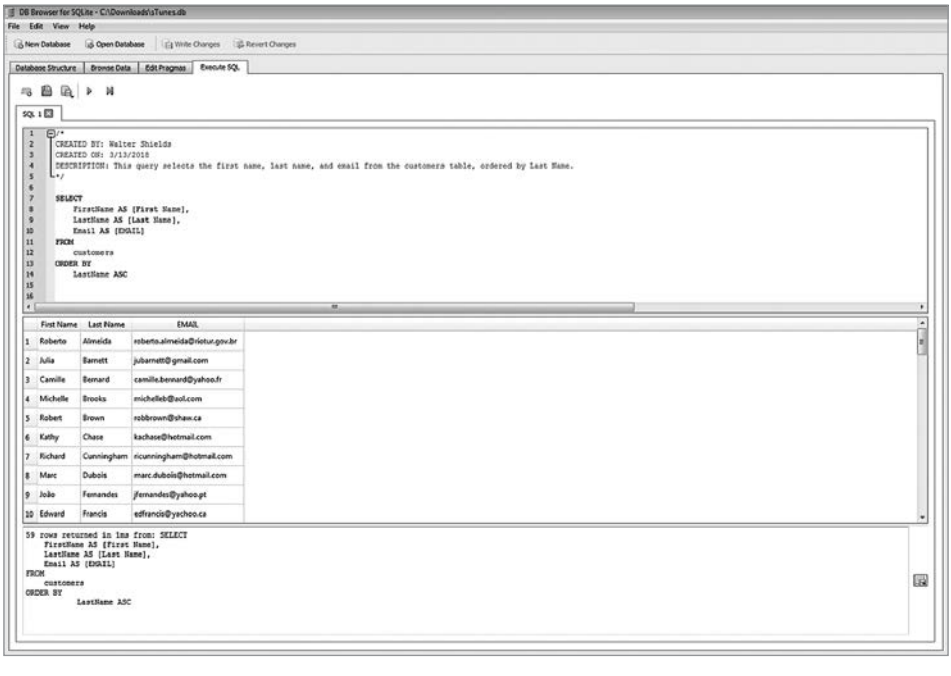

Рис. 37

ПРИМЕЧАНИЕ

Если условие ORDER BY отсутствует, каждый запрос будет возвращать данные в том порядке, в котором они были изначально сохранены в таблице.

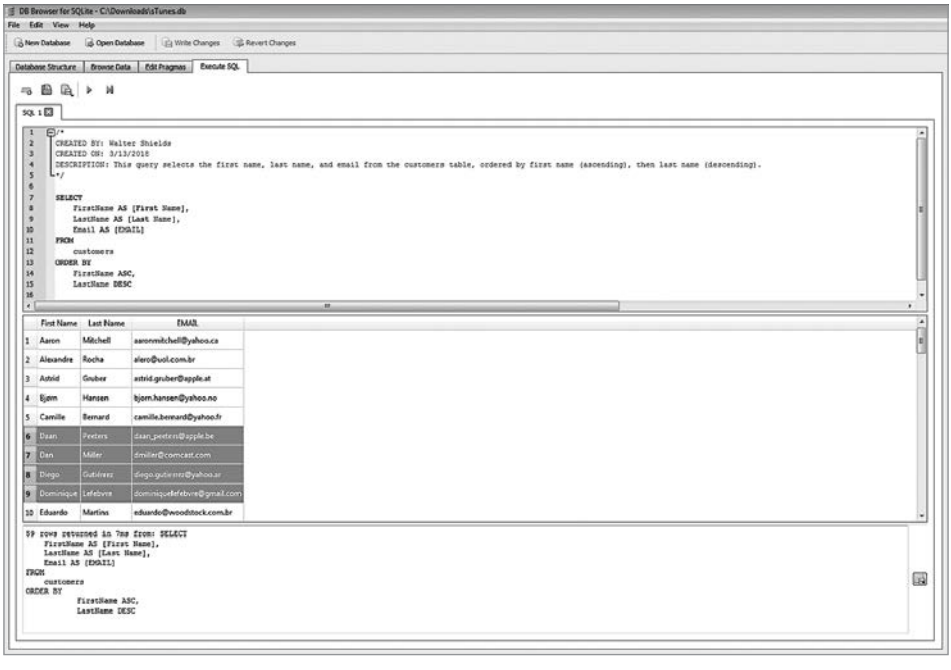

Рис. 38

Также условие `ORDER BY` мы можем использовать для сортировки по нескольким столбцам. В данном примере мы выполним сначала сортировку по имени (по алфавиту), а затем по фамилии (в обратном порядке). Для этого нам необходимо в условии `ORDER BY` указать два поля. При перечислении нескольких полей их необходимо разделять запятыми.

```
/*
CREATED BY: Уолтер Шилдс
CREATED ON: 13.03.2018
DESCRIPTION: Данный запрос осуществляет выборку полей имени, фамилии
и электронной почты из таблицы customers (клиенты), отсортированных
сначала по имени (по возрастанию), а затем по фамилии (по убыванию).
*/

SELECT
   FirstName AS [First Name],
   LastName AS [Last Name],
   Email AS [EMAIL]
FROM
   customers
```

```
ORDER BY
   FirstName ASC,
   LastName DESC
```

Выполним данный запрос и проанализируем имена клиентов, начинающиеся с буквы *D*. Мы видим, что имена расположены по алфавиту, а фамилии — в обратном порядке.

ПРИМЕЧАНИЕ

Если вы используете условие ORDER BY для столбца, в котором имеются пустые строки, вы увидите, что эти значения будут отображаться в начале списка как NULL (если сортировка осуществляется в порядке возрастания).

Практические задания

✶ Измените порядок полей в условии `SELECT` и установите `LastName` первым столбцом, а не вторым. Выполните запрос, в результате которого будут получены упорядоченные записи `LastName`. Стал ли список более читабельным?

Получение ограниченного числа записей с помощью условия LIMIT

В предыдущих примерах мы осуществляли выборку всех записей из таблицы `customers`. Хотя мы ограничили наши записи тремя полями и отсортировали эти поля, в панели сообщений мы видим, что каждый раз мы возвращаем 59 строк. Если нет необходимости в просмотре всех 59 записей, мы можем ограничить наши результаты определенным количеством строк. Это бывает полезно при сортировке по количеству (которое мы продемонстрируем позже), например по самой высокой цене или самым большим продажам. Если после условия `ORDER BY` мы добавим ключевые слова `LIMIT 10`, в результате вернутся только первые десять записей в указанном порядке сортировки. Можно указать любое число записей, которое необходимо отобразить (при условии, что в таблице имеется такое количество записей).

```
/*
CREATED BY: Уолтер Шилдс
CREATED ON: 13.03.2018
DESCRIPTION: Данный запрос осуществляет выборку первых 10 записей из
таблицы customers (клиентов), отсортированных сначала по имени (по
возрастанию), а затем по фамилии (по убыванию).
*/
```