

Интернет-журнал «Мир науки» / World of Science. Pedagogy and psychology <https://mir-nauki.com>

2018, №6, Том 6 / 2018, No 6, Vol 6 <https://mir-nauki.com/issue-6-2018.html>

URL статьи: <https://mir-nauki.com/PDF/67PDMN618.pdf>

Статья поступила в редакцию 24.11.2018; опубликована 21.01.2019

**Ссылка для цитирования этой статьи:**

Пирогов В.Ю. Некоторые особенности преподавания языка управления базами данных // Интернет-журнал «Мир науки», 2018 №6, <https://mir-nauki.com/PDF/67PDMN618.pdf> (доступ свободный). Загл. с экрана. Яз. рус., англ.

**For citation:**

Pirogov V.Ju. (2018). Some features of teaching database management language. *World of Science. Pedagogy and psychology*, [online] 6(6). Available at: <https://mir-nauki.com/PDF/67PDMN618.pdf> (in Russian)

**УДК 004.5**

**ГРНТИ 14.35.09**

**Пирогов Владислав Юрьевич**

ФГБОУ ВО «Шадринский государственный педагогический университет», Шадринск, Россия  
Заведующий кафедрой «Программирования и автоматизации бизнес-процессов»  
Кандидат физико-математических наук, доцент  
E-mail: [Vladislav-133@yandex.ru](mailto:Vladislav-133@yandex.ru)

## **Некоторые особенности преподавания языка управления базами данных**

**Аннотация.** Статья посвящена вопросам преподавания языка SQL. Язык SQL является декларативным языком, т. е. в нем описываются параметры результатов или действий, но не описывается алгоритм достижения цели. Подчеркивается, что язык SQL реляционный и ориентирован на взаимодействие с реляционными базами данных. Рассматриваются особенности команд языка SQL на примере команды SELECT – извлечения информации из реляционной базы данных. Кратко рассматривается методика, позволяющая облегчить учащимся формирование структуру команды SELECT. Отмечается, что использование визуальных методов и языка «Запросов по образцу» (QBE) нецелесообразно, поскольку данный язык не используется в профессиональном программировании и не позволяет продвинуться в использовании языка SQL. Целесообразнее более углубленно изучать язык SQL. Авторы указывают, что для освоения языка SQL требуется использование автоматизированной системы, которая должна удовлетворять следующим требованиям: 1. Возможность использовать как в аудитории, так вне ее. 2. Система должна позволять: а) несколько попыток выполнения; б) просмотр результата; в) просмотр эталонного запроса. 3. Контроль со стороны преподавателя, в том числе и в онлайн режиме. 4. Автоматический вывод результатов по каждой теме. В качестве критерия проверки осуществляется сравнение эталонного набора с результирующим набором, полученным из написанного студентам запроса. Далее авторы описывают автоматизированную систему, позволяющую осуществлять интерактивное выполнение студентами запросов, с контролем их выполнения. Данная система функционирует на сервере и может быть использована студентами через web-интерфейс. Для аудиторного применения была также разработано оконное приложение, используемая в настоящее время в тестовом режиме.

**Ключевые слова:** запрос; декларативные языки; императивные языки; базы данных; реляционный; автоматизированная система; связи между таблицами; первичный ключ

## Введение

Язык SQL (Structured Query Language – Язык Структурированных Запросов) является основным языком взаимодействия с реляционными базами данных. Написание запросов на языке управления реляционными базами данных является составным элементом многих дисциплин: Базы данных, Информационные системы, Проектирование информационных систем, Web-программирование и многих других. Команды языка SQL входят в большинство современных информационных систем.

Следует отметить еще один важный аспект использования языка SQL в практике работы с базами данных. Очень часто разработчики информационных систем не успевают за потребностями конечного пользователя. В этом случае для получения нужных отчетов прикладным программистам и администраторам баз данных приходится использовать данный язык непосредственно, как говорят интерактивно. Иногда это становится проблемой, поскольку приходится вводить в штат специалиста, владеющего именно языком SQL.

Классический язык SQL [1, 2] не является языком программирования в традиционном смысле – алгоритмическим языком. Он всегда выступает в качестве инструмента, осуществляющего интерфейс с реляционными базами данных. Однако создатели этого языка заложили в него колоссальные возможности, позволяющие осуществлять очень сложные манипуляции с данными, в частности возможность извлекать информацию из многих таблицы одновременно, проводя отбор по комплексному набору параметров, выстраивая иерархию запросов. Умение писать запросы на языке SQL является важнейшей компетенцией практически для любого разработчика прикладного программного обеспечения и администратора баз данных.

Особенностью языка SQL является то, что он декларативен [3-5]. Другими словами, в нем задается описание (спецификация) конечного результата, но не описывается последовательность действия, которая приводит к необходимому результату. Если учесть, что в рамках курсов, которые в основном читаются для современных технических специальностей, рассматриваются в основном императивные языки [4], изучение такого языка, как SQL приобретает самостоятельную ценность.

Основы языка SQL описаны во многих книгах [1-3, 6], но, как и при обучении программированию, при подготовке будущего специалиста, требуется практика, обучаемый должен «набить» руку. Конечно, при написании прикладных программ в рамках курсовых работ или проектов в рамках изучаемой дисциплины, студенту приходится так или иначе писать запросы на языке SQL. Однако такие запросы часто диктуются требованиями проекта, а не необходимостью изучения самого языка. Нужно учесть и еще один важный момент, характеризующий современные процессы в области высшего образования – увеличение часов на самостоятельную внеаудиторную работу студентов. Необходим некоторый механизм, позволяющий с одной стороны студенту практиковаться в изучении языка, с другой стороны сохранить контроль над внеаудиторной работой студента со стороны преподавателя. Такой контроль возможен при наличии автоматизированной системы.

Отдельно следует остановиться на вопросах преподавания языка SQL в средней школе. На наш взгляд школьные программы и учебники [7-9] не достаточно уделяют внимание этой теме, описывая в основном некоторые инструменты визуального формирования запросов, что не позволяет часто научиться написанию даже простейших запросов.

В рамках данной статьи мы рассмотрим некоторые вопросы преподавания языка SQL, а также вопросы автоматизации подготовки студентов по данной теме. Последнее, на наш взгляд, является важным элементом в подготовке будущих специалистов в области разработки прикладного программного обеспечения.

Таким образом актуальность рассматриваемой темы обусловлена: 1. Широким использованием языка SQL в современном программном обеспечении. Применением языка помимо прикладного программного обеспечения и в интерактивной форме для получения актуальной информации, доступ к которой на данный момент не возможен через стандартное ПО. 2. Включением данного языка в программы подготовки специалистов в области прикладного программного обучения, программы подготовки учителей информатики, школьные курсы информатики. 3. Увеличением часов на самостоятельную внеаудиторную работу студентов и необходимостью определенного контроля над такой работой со стороны преподавателя при помощи автоматизированной системы.

### Язык SQL

Язык SQL ориентирован на конкретную модель данных – реляционную [6, 10-14]. Эта модель возникла на рубеже 1970-х в одной из лабораторий корпорации IBM [15-18]. Структура оказалась не только относительно простой, но и гибкой, так что в последствие большая часть баз данных создавалась именно на основе этой модели. В последствие для реляционной модели был разработан язык – SQL.

Реляционная (табличная) база данных (см. [19, 20]) состоит из таблиц, связанных друг с другом посредством первичных и внешних ключей [6, 12]. Можно говорить о том, что реляционная база данных представляет собой отображение сущностного описания предметной области в табличную область<sup>1</sup>.

Как было указано выше язык SQL является декларативным. Это значит, что команды этого языка описывают некоторый конечный результат: описание структуры таблицы, которая должна быть создана; описание набора данных, который должен быть извлечен из реляционной базы данных; описание столбцов и строк таблицы, изменения в которых должны быть осуществлены. В командах отсутствуют какие-либо указания, как производить извлечение и сортировку результирующих строк, как осуществлять обновление или удаление строк таблицы и т. п. Другими словами, какой алгоритм для этого использовать. Последнее возлагается на исполняемую систему – СУБД (Система Управления Базами Данных). Это совсем нетривиальная вещь, поскольку программисты, занимающиеся разработкой на императивном языке, должны описывать не результат, а последовательность шагов, приводящая к заданному результату<sup>2</sup>. Сам конечный результат может быть достаточно сложным образом связан с исходной структурой данных и программист, таким образом, может сосредоточиться на правильном описании этих связей и не думать об алгоритме реализации.

Важно подчеркнуть, что язык SQL стандартизованный язык, что частично объясняет его превалирование над другими реляционными языками. И хотя стандарты не требуют полного их соблюдения для разработчиков СУБД, они позволяют программистам создавать переносимые приложения. Ядро языка одинаково выполняется на разных СУБД, что позволяет использовать в учебном процессе любую из реляционных СУБД.

---

<sup>1</sup> Упрощенно можно говорить, что сущность отображается на таблицу.

<sup>2</sup> Уже практически не используемая СУБД FoxPro имела язык управления со встроенным языком SQL и одновременно позволяла писать императивные (пошаговые) алгоритмы обработки в реляционных базах данных [6].

Множество команд языка SQL обычно делят на три подмножества: DML, DDL, DCL<sup>3</sup>. Как правило, второе и третье подмножества используются редко, в основном при проектировании базы данных и в дальнейшем это обычно прерогатива администраторов баз данных. Первое же подмножество составляет основу интерфейса с данными и используется в программировании информационных систем. Знание этого подмножества – основа профессиональных компетенций будущих программистов. Подмножество DML состоит из четырех команд. Три команды редактирования данных: добавление записей, удаление записей, изменение содержимого записей<sup>4</sup>. Данные команды достаточно просты и никогда не вызывают трудности у обучаемых. Четвертая команда SELECT используется для извлечения информации из базы данных по заданным критериям. Обладает сложной структурой и возможностью наращивания функциональности.

### Особенности преподавание языка SQL

Есть смысл говорить об языке SQL только в тесной увязке с принципами построения реляционных баз данных. Стандарты языка SQL содержат в себе и описание реляционных принципов [6, 10]<sup>5</sup>. На наш взгляд преподавание реляционных баз данных должно происходить вместе с изучением реляционного языка. Например, структура и свойства таблицы и команда SELECT; связи между таблицами – многотабличные запросы, позволяющие объединять таблицы как по первичным-внешним ключам, так и на основе динамической связи; типы данных в реляционных базах данных – типы данных из стандарта SQL и т. д. Студенты должны сразу понимать, что вся структура и содержимое базы данных может быть описано последовательностью команд SQL.

Очень часто, особенно в школьных курсах информатики [7-9], преподавание основ SQL заменяется использованием языка QBE<sup>6</sup>. Данный язык представляет собой простой визуальный способ формирования запросов к реляционным базам данных. Обычно все инструменты, предоставляющие возможность формировать запрос на языке QBE, дают соответствующий эквивалент на языке SQL. Язык действительно прост, по этой причине он привлекает и учителей, и преподавателей Вузов. В действительности из-за данной простоты обучаемый попадает в ловушку. С одной стороны, язык SQL оказывается не освоенным, с другой стороны язык QBE не позволяет формировать сложные запросы, например, с подзапросами или объединениями запросов. Т. е. QBE абсолютно бесполезен в профессиональной деятельности. Вопрос: «зачем вообще его нужно преподавать?» – повисает в воздухе. Увлеченность изучением визуальных инструментов, в ущерб постижению концептуальных и алгоритмических смыслов, к сожалению, не так уж редко встречается в преподавании информатики<sup>7</sup>. Конечно, визуализация может значительно помочь при работе со сложными программными системами. Но в данном случае речь идет о визуализации, как вспомогательном средстве, позволяющем охватить систему в целом, подметить связи и тонкие места,

<sup>3</sup> DML – Data Modification Language (язык модификации данных), DDL – Data Definition Language (язык описания данных), DCL – Data Control Language (язык управления данными).

<sup>4</sup> В действительности достаточно и первых двух команд: изменение содержимого строк можно свести к команде удаления и добавления.

<sup>5</sup> Стандарты языка SQL находятся в свободном доступе. См. например <http://www.contrib.andrew.cmu.edu/~shadow/sql/sql1992.txt>.

<sup>6</sup> QBE – Query by Example, т. е. запрос по образцу.

<sup>7</sup> Даже в учебнике Информатика. Углубленный курс. К.Ю. Полякова и Е.А. Еремина, где в главе Базы данных дается очень полное и корректное описание реляционных баз данных, вопрос о языке SQL излагается конспективно.

автоматизировать некоторые рутинные операции, а не как основное средство разработки. Сказанное выше в полной мере относится и к языку SQL. Существующие визуальные средства позволяют моделировать на нем довольно сложные запросы и могут быть использованы как вспомогательные инструменты, особенно в случае интерактивного использования языка.

При планировании преподавания языка SQL следует учитывать следующие факторы: 1. Структуру языка. Выше мы отметили, что язык можно представить в виде объединения трех множеств: DML, DDL, DCL. 2. Частоту использования команд. Команды DDL и DCL используются редко, поскольку структура базы данных задается обычно при проектировании, а параметры безопасности меняются довольно редко. И в этом случае штатные средства используемых визуальных инструментов вполне достаточны. Команды же множества DML используются непосредственно при разработке программного обеспечения информационных систем и в интерактивном режиме. Таким образом, если при изучении команд множеств DDL и DCL можно ограничиться их пониманием, то команды множества DML должны быть хорошо освоены практически. 3. Сложность команд. Как уже было отмечено из четырех команд множества DML, сложной является команда SELECT. В ее формате заложена возможность наращивать ее функциональность путем: а) соединения большего количества таблиц; б) объединения множества запросов в один; в) использования подзапросов; г) использования сложных логических условий. Еще одной важной особенностью команды SELECT является то, что она предназначена для формирования отчетов в зависимости от потребности конечного пользователя. Поэтому ее использование выходит за рамки чисто программного использования, а диктуется изменяющимися потребностями в получении той или иной информации.

Фактически все команды SQL с точки зрения преподавания можно представить в виде пирамиды. В вершине пирамиды располагается команда SELECT самая важная с практической точки зрения и требующая специальной подготовки. Ниже уровнем располагаются остальные команды множества DML. Эти команды часто используются, но их использование ограничивается в основном использованием в программном коде. И наконец еще ниже можно расположить все остальные команды, которые студент должен понимать и знать инструментарий их реализации.

### Автоматизация обучения языку SQL

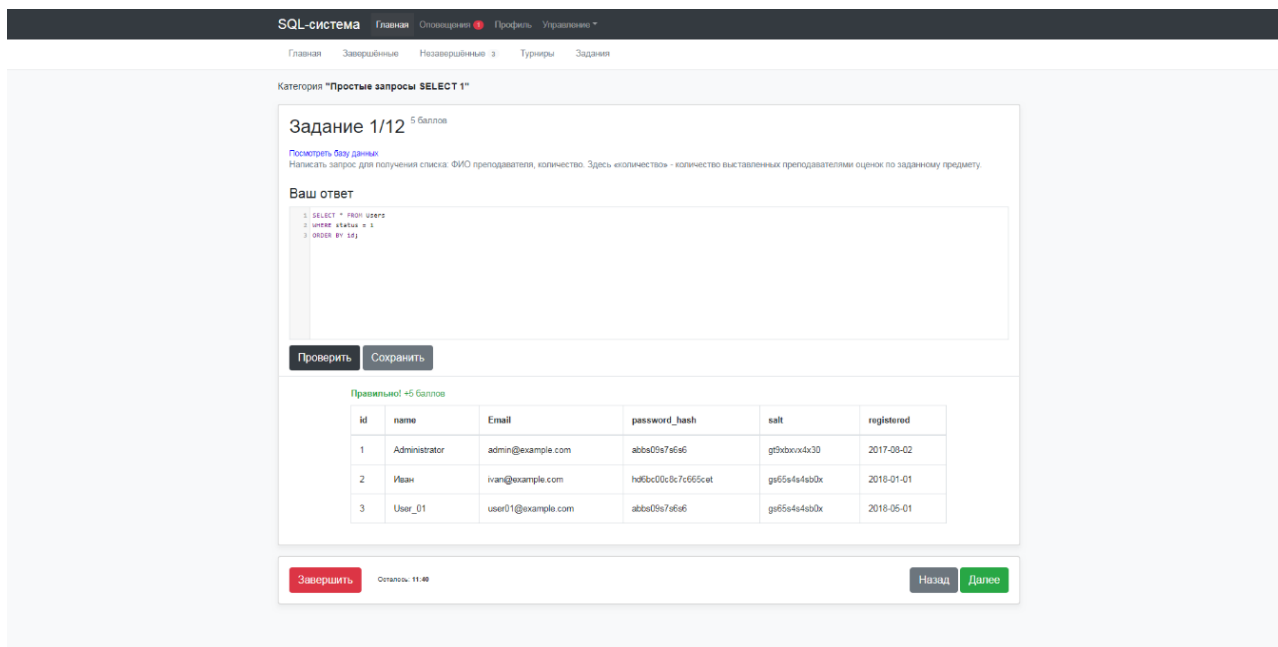
В предыдущем разделе было показано, что все множество команд можно представить в виде пирамиды. При этом для освоения нижней части пирамиды вполне достаточно аудиторных занятий. Команды средней части относительно просто и легко могли бы быть усвоены во время аудиторных занятий. Но поскольку они используются при разработке прикладного программного обеспечения (в проектах по дисциплине, курсовых проектах, ВКР) то их практическое освоение не встречает никаких особых трудностей. Но команда SELECT, расположенная в вершине пирамиды, требует специального подхода, поскольку она и рассчитана на использование ее для извлечения из базы данных самой разной информации. Естественно было бы вынести ее на внеаудиторное время. Однако, желательно при этом, чтобы студенты могли бы тренироваться вне аудиторий. Такую возможность могла бы обеспечить автоматизированная система. Подобная система должна обладать следующими возможностями:

1. Возможность работы с ней как в аудитории, так и вне ее. Такую возможность дает Интернет. Система должна быть централизованной.
2. Взаимодействие студента (учащегося) с ней должно позволять:
  - а) несколько попыток выполнения;

- б) возможность выполнения заданий в произвольном порядке;
  - в) просмотр результатов;
  - г) просмотр эталонного запроса, представленного преподавателем.
3. Контроль со стороны преподавателя за выполнение заданий, в том числе и в онлайн режиме: кто и сколько заданий выполнил и с какого количества попыток.
4. Автоматический вывод результатов по каждому студенту (таблица результатов).
5. Возможность формирования групповых и индивидуальных заданий, формирование групп студентов, тем, дисциплин.

Важный вопрос, который возникает при разработке таких систем: каков критерий проверки. Очевидно, что критерием проверки не может быть некая эталонная команда. Она полезна только для просмотра обучаемым после того, как он выполнил задание. Но в отличие от проверки обычных программ, правильность которых проверяется набором тестовых заданий, в данном случае можно ограничиться только одним правильным набором данных. Этот набор будет давать тем более объективный результат, чем большее количество данных будет содержаться в предложенной базе. Другими словами, вероятность того, что не правильный запрос может дать правильный результат будет стремиться к нулю. Задание может иметь, таким образом, несколько решений, каждое из которых будет считаться верным.

На факультете Информатики, математики и физики ФГБОУ ВО ШГПУ ведется разработка такой системы, отдельные части которой проходят бета-тестирование. Система представляет собой серверную часть (СУБД PostgreSQL) и несколько клиентских приложений.



*Рисунок 1. Внешний web-ориентированной автоматизированной системы проверки SQL-запросов*

На рисунке 1 представлен скриншот web-приложения, позволяющего осуществлять автоматическую проверку правильности команд SQL. Особенность приложения заключается в том, его можно использовать в любое время через сеть Интернет. Студенты могут пользоваться для работы с ней также и мобильными устройствами. В настоящее время сайт расположен по адресу <https://sql.ma-dev.cloud/> и проходит тестовые проверки.

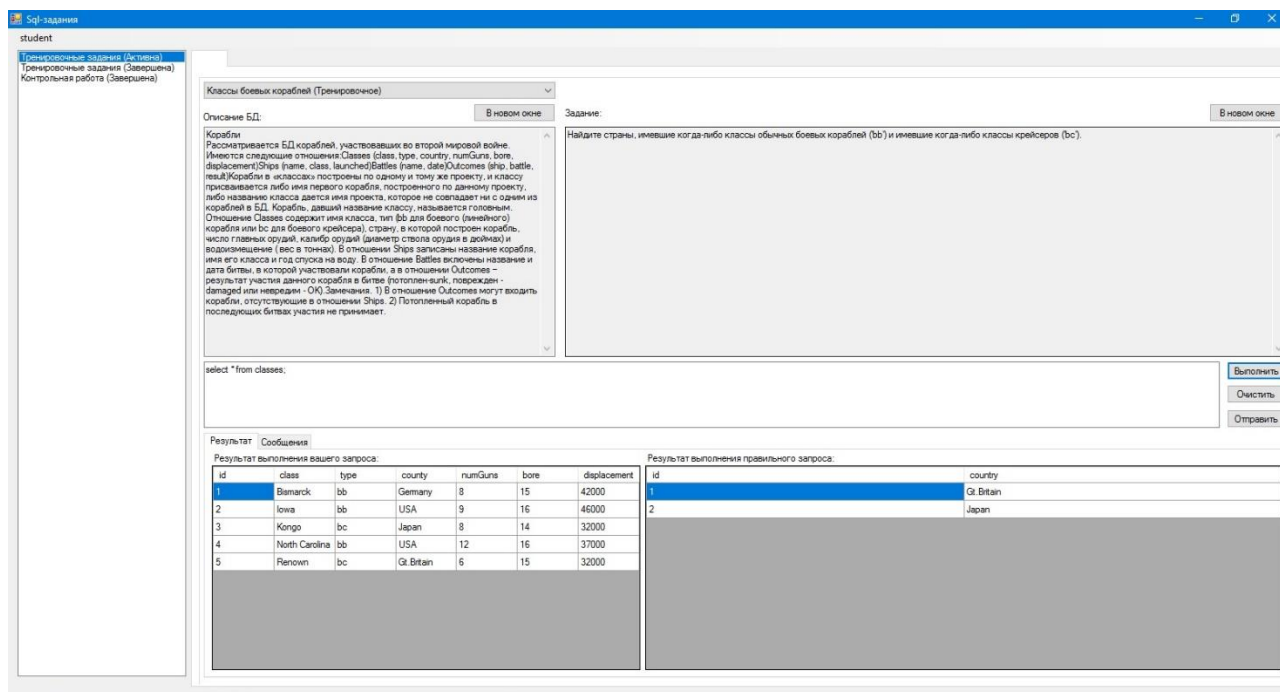


Рисунок 2. GUI-интерфейс программы автоматической проверки SQL-запросов

Ранее было разработано графическое приложение для Windows, которое рассчитано для работы в аудитории (см. рисунок 2). Приложение прошло государственную регистрацию (свидетельство №2017661010 от 02.10.2017). В настоящее время оно используется на занятиях. Наиболее сложной частью подобной системы является разработка административной части. В настоящее время ведется работа над ней в рамках грантовой поддержки.

И web-приложение и оконное приложение работают с одним и тем же виртуальным сервером. Очень важным, на наш взгляд, будет подробная методика использования системы, которая будет нарабатываться в процессе ее эксплуатации.

## Выводы

В статье были рассмотрены некоторые вопросы преподавания языка SQL. Отметим следующие важные моменты.

1. Умение использовать язык SQL в профессиональной деятельности – важнейшая профессиональная компетенция специалистов в области прикладного программирования и баз данных.
2. При изучении языка SQL следует структурировать его по таким параметрам как подмножества языка, частоту использования команд и сложность команд.
3. Особый упор в практической подготовке студентов следует сделать на команду SELECT.
4. Для успешного освоения данной командой необходима автоматизированная информационная система, позволяющая тренировать навыки написания команд во внеаудиторное время. Пример такой системы представлен в статье.

## ЛИТЕРАТУРА

1. Грабер, М. Введение в SQL [Текст] / М. Грабер. – М.: Лори, 2018. – 374 с.
2. Молиаро, Э. SQL. Сборник рецептов [Текст] / Э. Молиаро. – СПб.: Символ-Плюс, 2016. – 665 с.
3. Грофф, Джеймс. SQL. Полное руководство [Текст] / Джеймс Р. Грофф, Пол Н. Вайнберг. – Киев: ВНУ-Киев, 2001. – 960 с.
4. Van Roy, Piter. Concepts, Techniques, and Models of Computer Programming [Text] / Piter Van Roy, Seif Haridi. – England: The MIT Press Cambridge, Massachusetts London, 2004. – 900 p.
5. Kent, D. Lee. Foundations of Programming Languages [Text] / D. Lee. Kent. – Springer, 2015. – 234 p.
6. Пирогов, В.Ю. Информационные системы и базы данных [Текст] / В.Ю. Пирогов. – СПб.: БХВ, 2009. – 528 с.
7. Угринович, Н.Д. Информатика и ИКТ. Учебник для 11 класса. [Текст] / Н.Д. Угринович. – М.: Бином, 2009. – 314 с.
8. Семакин, И. Информатика 11 класс [Текст] / И. Семакин, Е. Хенер. – М.: Бином, 2005. – 144 с.
9. Поляков, К.Ю. Информатика 11. Углубленный уровень [Текст] / К.Ю. Поляков, Е.А. Еремин. – М.: Бином, 2013. – 241 с.
10. Дейт, К.Дж. Введение в системы баз данных [Текст] / К.Дж. Дейт. – 8-е изд. – М.: Вильямс, 2005. – 1328 с.
11. Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика [Текст] / Т. Коннолли, К. Бегг. – 3-е изд. – М.: Вильямс, 2003. – 1436 с.
12. Мирошниченко, Г. Реляционные базы данных: практические приемы оптимальных решений [Текст] / Г. Мирошниченко. – СПб.: БХВ, 2005. – 392 с.
13. Стружкин, Н. Базы данных. Проектирование [Текст]: учебник / Н. Стружин, В. Годин. – М.: Юрайт, 2017. – 478 с.
14. Ульман, Джеффри. Реляционные базы данных [Текст] / Джеффри Ульман, Дженнифер Уидом. – М.: Лори, 2014. – 384 с.
15. Codd, E.F. A Relational Model of Data for Large Shared Data Banks [Text] / E.F. Codd // IBM Research Laboratory, San Jose, California, February Communications of the ACM (CACM), 1970.
16. Codd, E.F. Relational. Completeness of Data Base Sublanguages [Text] / E.F. Codd // Data Base Systems, Courant Computer Science Symposia Series 6. – Englewood Cliffs, N.S.: Prentice-Hall, 1972.
17. Chen, C. The Systematic Approach in Teaching Database Applications [Text] / C. Chen, C. Ray.
18. Information Technology, Learning, and Performance Journal. – Vol. 22. – No. 1. – 2004. – P. 9-21.
19. Codd, E.F. The relation model for Database Model [Text] / E.F. Codd. – Addison-wesley Publishing Company, 1990. – 540 p.
20. Пирогов, В.Ю. О некоторых терминах в дисциплинах прикладной информатики [Текст] / В.Ю. Пирогов // Мир науки. – 2017. – Т. 5. – № 4.
21. Светлов, А.В. Особенности методики преподавания курса Базы данных для направления подготовки бакалавриата «Прикладная информатика» [Текст] / А.В. Светлов // Вестник ВолГУ. Серия 6. – 2011-2012. – Вып. 13.



**Pirogov Vladislav Jurievich**

Shadrinsk pedagogical university, Shadrinsk, Russian  
E-mail: Vladislav-133@yandex.ru

## Some features of teaching database management language

**Abstract.** The article deals with the teaching of the SQL language. SQL is a declarative language, that is, it describes the parameters of results or actions, but does not describe the algorithm for achieving the goal. It is emphasized that the SQL language is relational and is focused on interaction with relational databases. The article discusses the features of SQL commands on the example of the command "SELECT" (extracting information from a relational database). The paper deals with a technique that allows students to facilitate the formation of the structure of the SELECT command. The authors note that the use of visual methods and the language "Query by Example" (QBE) is impractical, because this language is not used in professional programming and does not allow to advance in the use of SQL. It is more expedient to study in depth the SQL language. The authors point out that the development of SQL requires the use of a computerized system that must meet the following requirements: 1. The opportunity to use it both in a lecture hall and outside it. 2. The system must be able: a) to make multiple attempts; b) to view results; c) to view the standard query. 3. Control by the teacher, including on-line. 4. Automatic output of results for each topic. As a test criterion, the reference set is compared with the result set obtained from the query written by the students. Next, the authors describe a computerized system allowing students to perform interactive queries with control of their implementation. This system operates on the server and can be used by students through the web-interface. For classroom use was also developed window application, currently used in test mode.

**Keywords:** query; declarative languages; imperative languages; data bases; relational; automated system; relationships between tables; primary key