

# ПОВЫШЕНИЕ ОБЪЕКТИВНОСТИ ОЦЕНКИ ЗНАНИЙ СТУДЕНТОВ ПРИ ПРОВЕРКЕ ПРАКТИЧЕСКИХ ЗНАНИЙ ПО ДИСЦИПЛИНЕ БАЗЫ ДАННЫХ

Пак В.С. Email: Pak1137@scientifictext.ru

Пак Виталий Станиславович – старший преподаватель,  
кафедра информационных технологий,  
Ташкентский университет информационных технологий им. Мухаммада Ал-Хорезми,  
г. Ташкент, Республика Узбекистан

**Аннотация:** в статье рассмотрены основные проблемы объективности проверки знаний, возникающие при проверке знаний по языку SQL. Рассмотрены причины, которые не позволяют использовать тестирование в качестве объективного метода оценки знаний. В исследовании описаны 3 метода решений, приведен анализ преимуществ и недостатков каждого из этих решений. Описана реализация подхода генерации «на лету», как наиболее предпочтительного решения, в соответствии с его достоинствами. Определен список дальнейших работ по совершенствованию системы.

**Ключевые слова:** методы проверки знаний, базы данных, объективность, SQL, реляционная алгебра, множество.

## THE INCREASING OF OBJECTIVITY OF ASSESSING STUDENTS' KNOWLEDGE WHILE PRACTICAL KNOWLEDGE TESTING IN THE DISCIPLINE OF THE DATABASE

Pak V.S.

Pak Vitaliy Stanislavovich – Associate Lecturer,  
INFORMATION TECHNOLOGIES DEPARTMENT,  
TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES NAMED AFTER MUKHAMMAD AL-KHOREZMI,  
TASHKENT, REPUBLIC OF UZBEKISTAN

**Abstract:** the article discuss the main problems of objectivity of knowledge verification, which arise when checking knowledge of the SQL language. The reasons that do not allow testing to be used as an objective method of knowledge assessment are considered. The study describes three methods of solutions and analysis of the advantages and disadvantages of each of these solutions. The realization of the approach "generation on the fly" is described as the most preferable solution, in accordance with its merits. A list of further works for the improvement of the system has been determined.

**Keywords:** methods of knowledge checking, databases, objectivity, sql, relational algebra, set.

УДК 378.146

### 1. Введение.

При подготовке специалистов по компьютерному инжинирингу огромное внимание уделяется формированию навыков работы с реляционными базами данных. Знание реляционного языка запросов SQL просто необходимо для любого специалиста, начиная от системных и сетевых администраторов, заканчивая архитекторами приложений и программными инженерами.

Без проверки и контроля знаний студентов учебный процесс теряет смысл, т.к. отсутствие их не позволяет судить о степени усвояемости обучаемих материала той или иной дисциплины, а в конечном счете невозможно квалифицировать студента после окончания срока обучения.

### 2. Проблемы автоматизированного контроля знаний.

В настоящее время благодаря стремительному развитию информационных технологий компьютеры все чаще используются для контроля успеваемости студентов. Не является исключением дисциплина базы данных. В ней все чаще и чаще применяются средства автоматизированной проверки знаний в форме тестирования.

Преимущество тестирования - более полный охват материала проверяемой дисциплины, а недостаток - жесткие рамки ответа на вопросы, исключение рассуждений, дополнений, элементов творчества [1].

Что касается проверки теоретических знаний, то тестирование справляется просто отлично. Однако при проверке знания запросов языка SQL все немного сложнее. Дело в том, что язык SQL основывается на реляционной алгебре, которая позволяет добиться нужного результата несколькими способами. Рассмотрим пример запроса на языке SQL согласно схеме, приведенной на рисунке 1.1 (b). Нам необходимо вывести всех продавцов из таблицы sellers, которые из одного и того же города, что и их заказчики customer.

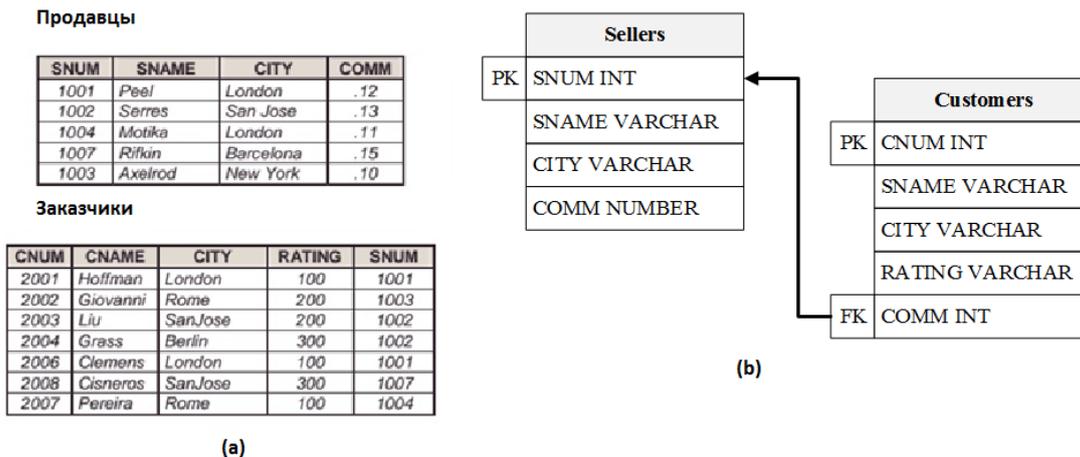


Рис 1.1. Пример схемы базы данных

Ответ на данное задание можно получить следующим образом, используя операцию соединения:  
 SELECT Sellers.\* FROM Sellers INNER JOIN Customers ON Sellers.SNUM = Customers.SNUM WHERE Sellers.CITY = Customers.CITY.

Или с помощью подзапроса:  
 SELECT \* FROM Sellers WHERE SNUM in (SELECT SNUM FROM Customers WHERE Customers.CITY = Sellers.CITY)

Существуют и другие варианты получения данного ответа. Все дело в реляционной алгебре, позволяющей получить один и тот же результат с помощью применения различных наборов операторов.

Данная возможность делает язык SQL достаточно мощным и универсальным. Выбор операторов для получения результата при этом становится делом вкуса любого из разработчиков базы данных или же определяется требованиями к производительности. Однако такая универсальность делает попытку автоматизации проверки знаний с помощью тестирования полностью неприемлемой. Поскольку для тестирования имеется всего один правильный вариант ответа (или несколько вариантов), а как было показано выше на практике существуют несколько возможностей получения одного и того же результата, то использование тестирования перестает быть объективным. Так как правильным ответом будет считаться тот, который заранее определил составитель теста, предпочитаемым для него набором реляционных операторов.

### 3. Подходы к решению.

Для решения проблемы предлагается использование системы, позволяющей сравнивать результаты «на лету». При использовании данного подхода сначала создается структура базы данных, на которой будут проверяться практические задания на языке SQL. После этого производится наполнение созданной базы данных строками. Затем подготавливается список вопросов и правильных ответов. Для того, чтобы дать возможность проверить ответ только на основе результата, возможно использовать несколько решений:

1. Предварительно хранить результат выполнения команды, т.е. для каждого из запросов вместо ответа в форме SQL необходимо хранить строки, соответствующие правильному ответу.
2. Можно произвести анализ на основе операций реляционной алгебры, позволяющий приводить запрос к формальной записи реляционных операторов и сравнить с эталонной формулой, дающей правильный ответ.
3. Можно хранить запрос, дающий правильный результат.

Первый способ является трудоемким, но обладает высокой надежностью. Недостатком данного способа является то, что при достаточно большом количестве строк, каждый вопрос будет требовать большого объема жесткого диска. Помимо этого, для разных запросов количество столбцов может быть разным, что требует создание специальной структуры для хранения результатов. Такой структурой может быть популярный на сегодняшний день JSON или XML. Однако все равно база данных, на которой производится контроль, будет статична, т.е. при изменении данных, добавлении, удалении и редактировании придется каждый раз производить обновление сохраненных результатов ответа.

Другой способ использования анализа запросов и построения из них формальной записи операторов (таблица 1.1), описанный Дж. Ульманом [3].

Таблица 1.1. Формальная запись операторов

Оператор	Формальная запись
выборка	$\sigma_{\text{predicate}}(\mathbf{R})$
проекция	$\Pi_{a_1, \dots, a_n}(\mathbf{R})$
объединение	$\mathbf{R} \cup \mathbf{S}$
вычитание (разность)	$\mathbf{R} - \mathbf{S}$
пересечение	$\mathbf{R} \cap \mathbf{S}$
декартово произведение	$\mathbf{R} \times \mathbf{S}$
соединение	$\mathbf{R} \bowtie_{\mathbf{F}} \mathbf{S}$
деление	$\mathbf{R} \div \mathbf{S}$

После преобразования запроса в формальное представление необходимо учесть комбинации согласно свойствам ассоциативности и коммутативности. После этого можно применить операторы преобразования. Данный подход является очень гибким, поскольку позволяет найти все возможные ответы и дает возможность динамично изменять структуру базы данных, на которой производится контроль. Однако данный подход является трудно реализуемым и требовательным к вычислительным мощностям.

Ввиду недостатков перечисленных предыдущих решений было принято решение остановиться на варианте хранения запроса, дающего правильный ответ. Была разработана таблица, в которой помещается задание и ответ на языке SQL. Для заданий устанавливается допустимое время на ее решение. Затем была разработана система для проведения контроля на языке JAVA с применением веб технологий.

Работа системы описывается последовательностью действий:

1. Для каждого из студента в случайном порядке формируется список заданий, затем друг за другом запускаются задания.
2. Запускается обратный отсчет, и студент в поле вводит запрос.
3. После нажатия кнопки отправки, система выполняет результат, отправленный студентом.
4. В случае если в синтаксисе запроса нет ошибок, извлекается правильный запрос из базы данных.
5. Он также выполняется и результат, полученный из запроса студента, сравнивается с результатом, полученным из запроса из базы.
6. В случае правильного ответа, производится переход к следующему заданию. Иначе переходит к пункту 2.

Данная система была успешно апробирована на итоговом контроле студентов 3-го курса направления компьютерный инжиниринг Ташкентского университета информационных технологий имени Муххамеда-Ал Хорезми. Использование системы показало хорошую производительность на тестовой базе, содержащей более 5 тыс. строк.

#### **Заключение.**

Апробация предложенного подхода и системы, реализующей ее, показала высокую объективность оценки знаний студентов и умение работать с языком SQL. Дальнейшие исследования предусматривают использование системы не только в процессе итогового или промежуточного контроля, но и как альтернативы для проверки усвоения знаний SQL по мере изучения отдельных разделов. Для этого требуется ввести учебный материал и разделить задания по разделам. После этого необходимо объединить систему проверки практических знаний с системой тестовых испытаний, что позволит охватить не только практическую, но и теоретическую часть.

#### ***Список литературы / References***

1. Слободин А.В., Часовских В.П. // Совершенствование оценки знаний методом тестирования. // Телематика 2002. Труды Всероссийской научно-методической конференции. СПб., 2002.
2. Connolly Tomas, Begg Carolyn // Database Systems 4th Ed. // Addison-Wesley, 2005. 1374 стр.
3. Уидом Дженифер, Ульман Джеффри Д. // Основы реляционных баз данных // Лори, 2006. 382.