

РЕФЕРАТ

НИР, с. 26, рис. 21, ист. 8, прил. 1

СИСТЕМА КОНТРОЛЯ ЗНАНИЙ, CODEWARS, ЗАДАНИЯ, СКРИПТЫ, SQL

Цель работы – выявить преимущества и недостатки одной или нескольких существующих систем контроля знаний языка запросов к базам для анализа наиболее важных составляющих интерфейса и функциональных возможностей продуктов и важность данных составляющих для системы, используемой для проведения собеседований и оценки знаний кандидатов на должность разработчиков баз данных.

Данная цель достигается посредством анализа существующих средств контроля знаний, а именно с помощью поиска информации в интернете и пробной их эксплуатации.

Существующие системы контроля знаний имеют свои преимущества и недостатки как с точки зрения тестируемого пользователя, так и с точки зрения разработчика тестовых заданий. Данная научно-исследовательская работа направлена на выявление преимуществ у существующих систем контроля знаний и последующей разработки технического задания на создание системы контроля знаний языка запросов к базе данных.

СОДЕРЖАНИЕ

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ	5
ВВЕДЕНИЕ	6
1 ВЫБОР ОБЪЕКТА АНАЛИЗА	7
2 АНАЛИЗ И ТЕСТИРОВАНИЕ ФУНКЦИОНАЛЬНОСТИ ДЛЯ КАНДИДАТА	8
2.1 Блок задания	9
2.2 Блок ввода решения	11
2.3 Блок результатов	12
2.4 Блок дополнительных функций	14
3 АНАЛИЗ И ТЕСТИРОВАНИЕ ФУНКЦИОНАЛЬНОСТИ ДЛЯ МЕНЕДЖЕРА	14
3.1 Блок управления	15
3.2 Блок информации о задании	16
3.3 Блок задания	17
3.4 Блок решения	19
3.5 Блок тестов	21
ЗАКЛЮЧЕНИЕ	24
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	26
ПРИЛОЖЕНИЕ А – ТЕХНИЧЕСКОЕ ЗАДАНИЕ	27

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

SQL (Structured Query Language) - декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.

IDE (Integrated Development Environment) — система программных средств, используемая программистами для разработки программного обеспечения.

ВВЕДЕНИЕ

В современных ИТ-компаниях проектные менеджеры достаточно большое количество времени тратят на собеседования, в том числе технические. Подготовка задания, его тестирование, проработка возможных ответов – это все занимает время менеджера, которое могло быть потрачено на развитие проектов компании.

Кроме того, техническая часть собеседования часто проводится на листочке, что дает только поверхностное понимание знаний кандидата, так как в рабочем процессе будет возможность использовать интернет ресурсы, собственные наработки – это все может значительно повысить производительность человека и некорректно лишать кандидата этих возможностей.

Данные проблемы могут быть решены использованием системы контроля знаний, которая позволит менеджерам создавать задания в онлайн форме, сохранять их, а затем, при необходимости, переиспользовать. Кандидату такая система позволит показать свои знания, используя все возможные источники информации, которые могут быть использованы в реальной работе.

1 Выбор объекта анализа

Существует большое количество систем, которые можно рассматривать, как системы контроля знаний. Тестирование может проводиться в различных форматах: от простых тестов до написания сложных скриптов вручную.

Среди русскоязычных ресурсов можно выделить ресурсы «geekbrains» и «proghub». Данные ресурсы позволяют пройти тестирование на знания как языка запросов, так и теоретических основ реляционных баз данных.

Так как оба ресурса являются аналогичными по функциональности, рассмотрим только один из них. На рисунке 1 представлен пример тестового задания с ресурса «Geekbrains».

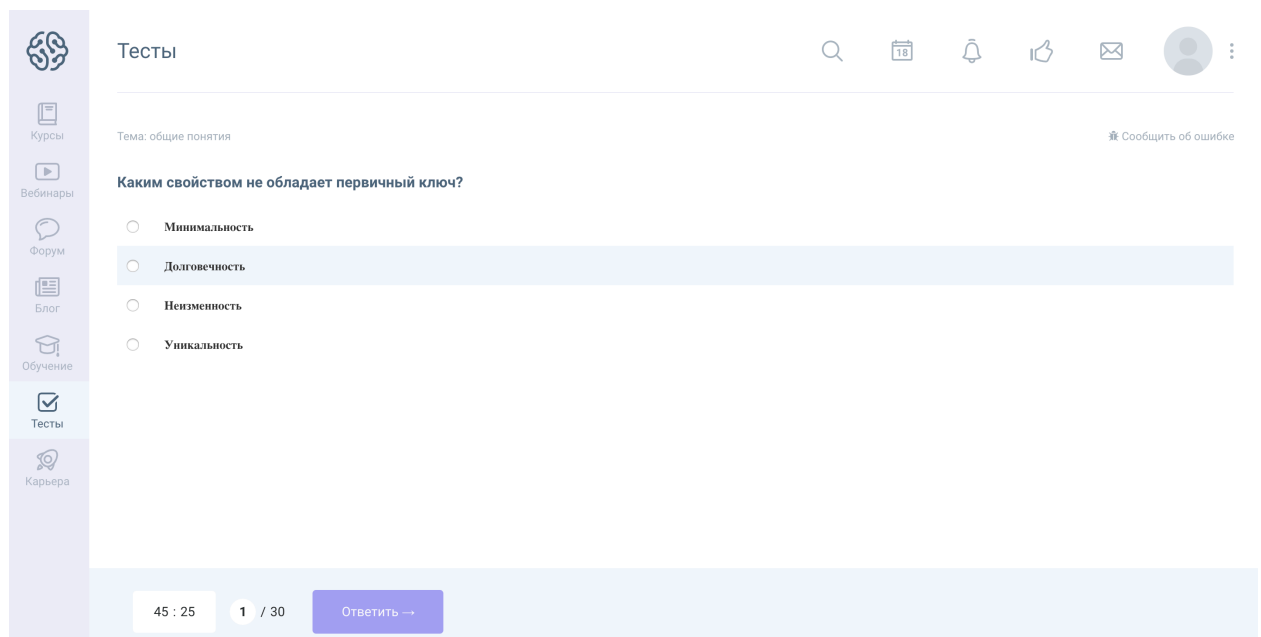
The image is a screenshot of a web interface for a test on the GeekBrains platform. On the left is a vertical sidebar with icons and labels for 'Курсы' (Courses), 'Вебинары' (Webinars), 'Форум' (Forum), 'Блог' (Blog), 'Обучение' (Education), 'Тесты' (Tests), and 'Карьера' (Career). The 'Тесты' section is currently active. The main content area is titled 'Тесты' and shows a question: 'Каким свойством не обладает первичный ключ?' (Which property does a primary key not have?). Below the question are four radio button options: 'Минимальность' (Minimality), 'Долговечность' (Durability), 'Неизменность' (Immutability), and 'Уникальность' (Uniqueness). The 'Долговечность' option is selected. At the bottom of the interface, there is a timer showing '45 : 25', a progress indicator '1 / 30', and a button labeled 'Ответить →' (Answer →).

Рисунок 1 – Пример тестового задания с «GeekBrains»

Данный портал позволяет проходить заранее подготовленный тест, который не может быть изменен при необходимости. Задания делятся по темам и могут быть как тестовыми, так и с развернутым ответом. Задания с развернутым ответом являются самыми простейшими и могут быть решены в одну строчку кода. По причине того, что данный ресурс не дает возможности изменять тесты и решать сложные задания с развернутым ответом, то он не подходит для анализа.

Среди англоязычных ресурсов можно выделить популярный портал «codewars». Данный портал используется для практики программирования и предоставляет возможности как для обучения программированию на различных языках, так и для проведения собеседований. Версия для проведения собеседований является платной и может быть использована только в коммерческих целях. По информации из открытых источников – функциональность версии для собеседований аналогична функциональности для обучения программирования. Следовательно, вся необходимая функциональность может быть протестирована в бесплатной версии. Данный портал имеет все необходимые для нас функции, поэтому для анализа будем использовать именно его.

2 Анализ и тестирование функциональности для кандидата

Типовое окно для тестируемого представлено на рисунке 2.

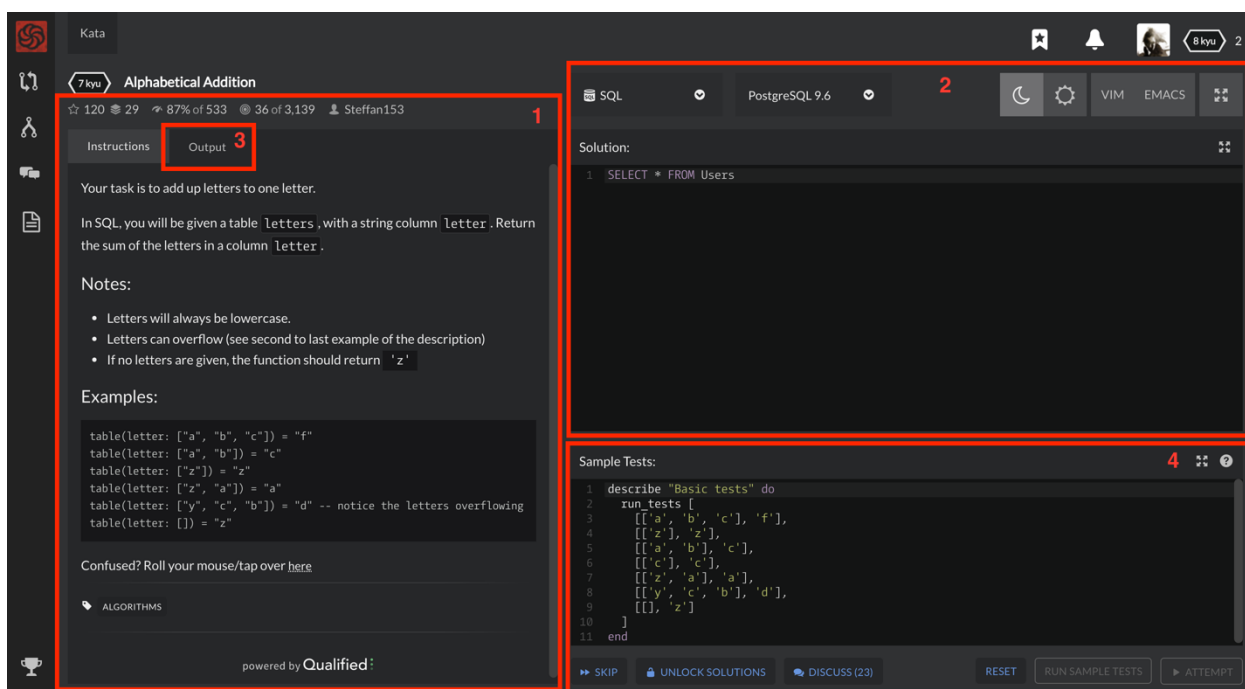


Рисунок 2 – Типовое окно тестируемого на портале «codewars»

Анализ «codewars» для кандидатов будет производиться по следующим пунктам:

- блок задания (1);
- блок ввода решения (2);
- блок результатов, который вызывается при запуске решения (3);
- блок дополнительных функций (4).

2.1 Блок задания

Блок задания представлен на рисунке 3.

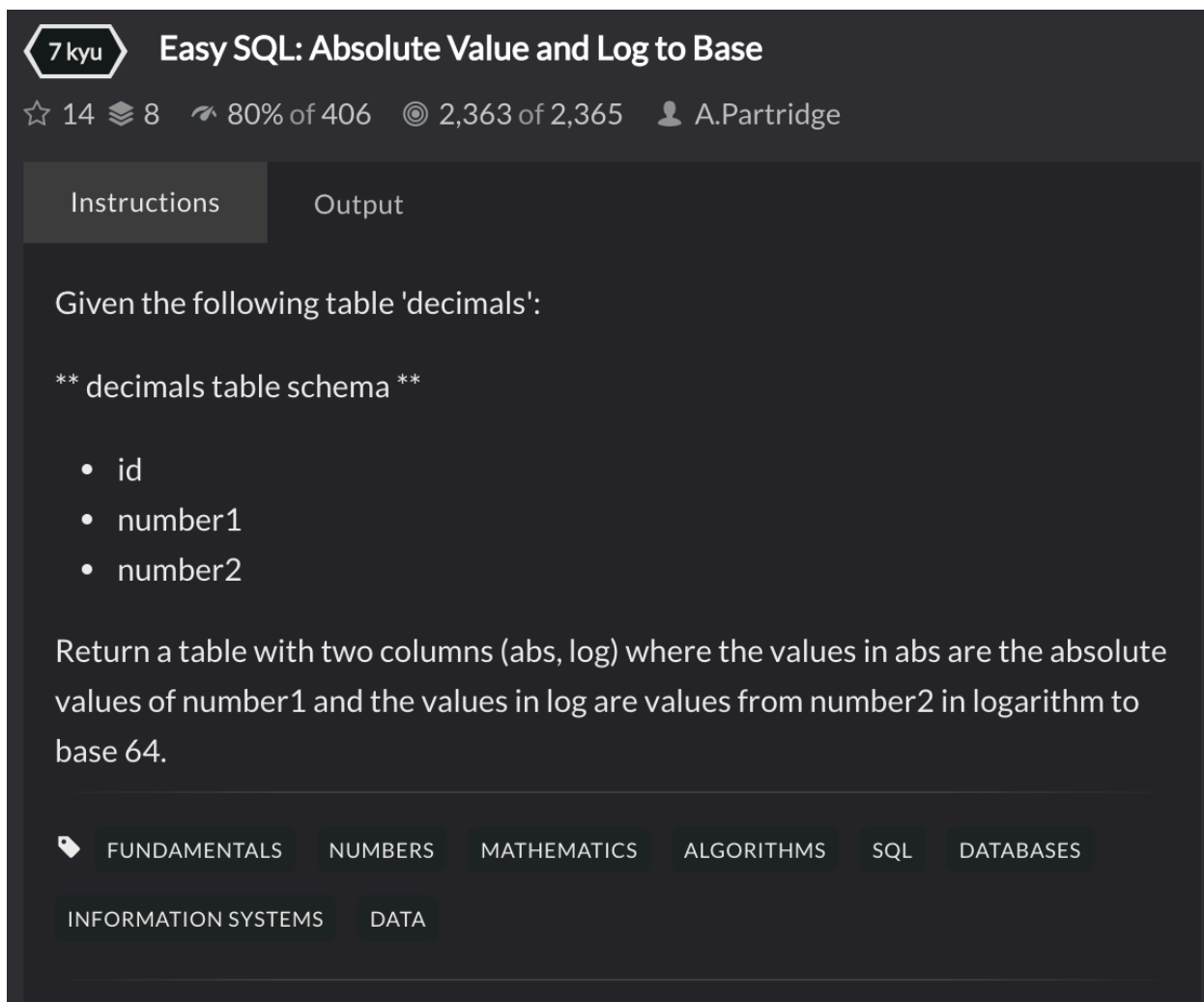


Рисунок 3 – Блок задания

Функционально данный блок можно разделить на 3 основные части: информация о задании, задание и теги задания. Рассмотрим каждую из них.

Информация о задании содержит:

- сколько раз задание добавлено в закладки,

- сколько раз задание добавлено в коллекции,
- пользовательский рейтинг,
- число успешных запусков,
- автор.

В ходе прочтения данной информации можно получить примерное представление о задании и о том, как его восприняло сообщество.

Самая большая часть в данном блоке отведена заданию. На примере, представленном на рисунке 3, задание представлено в текстовом виде и содержит информацию о рассматриваемом объекте и постановку вопроса. Задание может содержать особые выделенные структуры, которые хочет выделить автор (ключевые слова, примеры, куски кода). Пример с выделением таких структур представлен на рисунке 4.

Нижняя часть данного блока выделена для тегов задания. Теги могут быть использованы для сортировки задания (когда, например, пользователь хочет уделить особое внимание алгоритмам) и поиска.

Для проведения собеседований важными блоками будут автор и само задание. Информацию об удачных запусках, пользовательскому рейтингу задания, тегах, возможность добавлять в закладки и коллекции задание кандидату не нужно – это является избыточной для него информацией. Кроме того, для заданий на SQL было бы хорошо, если бы была возможность вставлять изображения в задания (например, схему БД).

Your task is to add up letters to one letter.

In SQL, you will be given a table `letters`, with a string column `letter`. Return the sum of the letters in a column `letter`.

Notes:

- Letters will always be lowercase.
- Letters can overflow (see second to last example of the description)
- If no letters are given, the function should return `'z'`

Examples:

```
table(letter: ["a", "b", "c"]) = "f"
table(letter: ["a", "b"]) = "c"
table(letter: ["z"]) = "z"
table(letter: ["z", "a"]) = "a"
table(letter: ["y", "c", "b"]) = "d" -- notice the letters overflowing
table(letter: []) = "z"
```

Confused? Roll your mouse/tap over [here](#)

Рисунок 4 – Пример задания с выделенными структурами

2.2 Блок ввода решения

Пример блока ввода решения представлен на рисунке 5

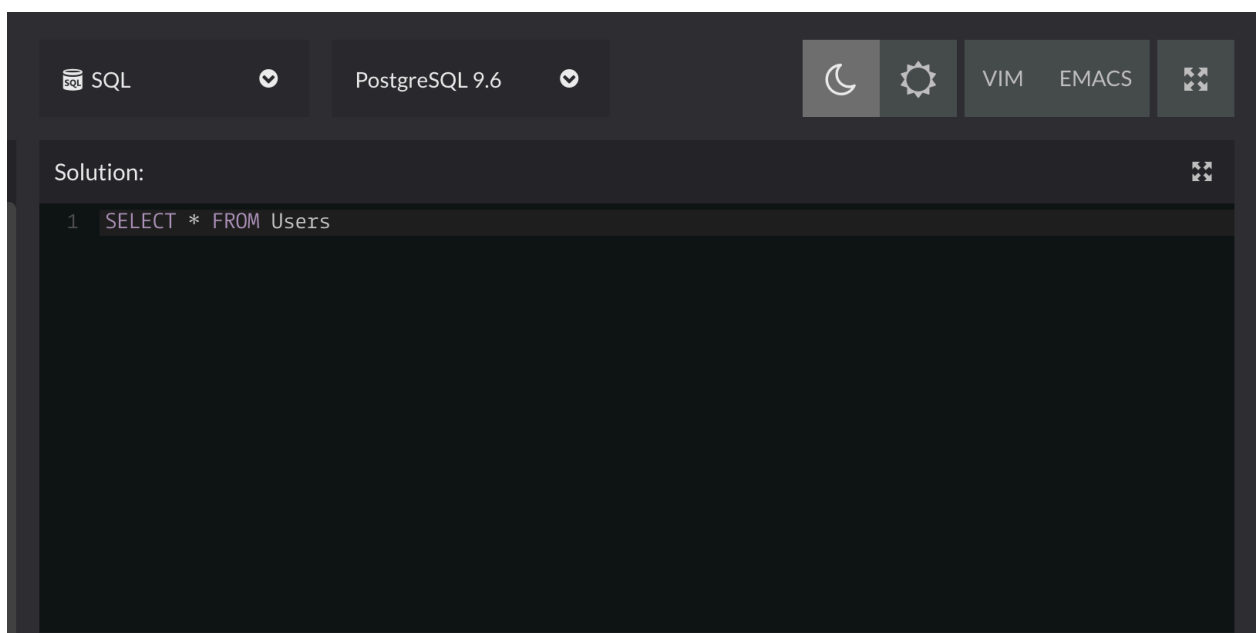


Рисунок 5 – Блок ввода решения

Данный блок содержит информацию о выбранном языке программирования и его версии (в данном случае – SQL, PostgreSQL 9.6). Это дает пользователю возможность выбрать предпочтительную версию/расширения языка для работы. В случае SQL пользователь может выбрать с какой СУБД работать. Данный выбор представлен на рисунке 6.

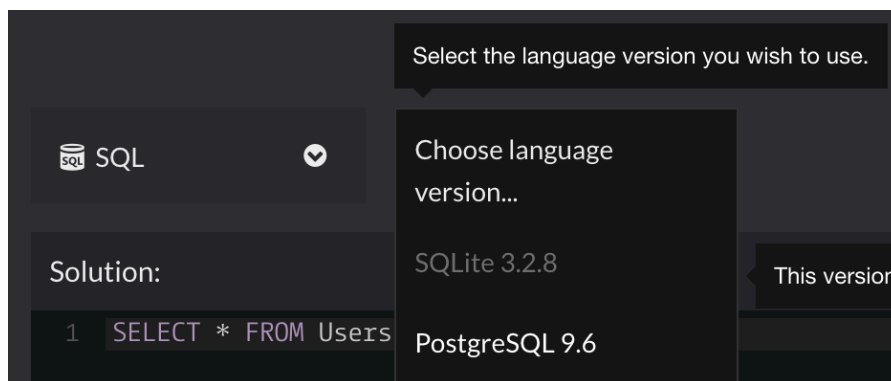


Рисунок 6 – Выбор версии языка программирования

Далее в данном блоке присутствует возможность выбора темы интерфейса (темная или светлая), выбора типа редактора кода (обычный, vim, emacs) и включение полноэкранного режима.

Основную часть данного блока занимает окно ввода кода. Данное окно представлено в виде простейшей IDE, которая подсвечивает служебные слова, нумерует строки, но не имеет более сложных функций типа помощь при вводе и подсветки неправильного синтаксиса.

Для собеседований необходимыми являются функции выбора версии языка (для каких-то заданий изменение версии языка может быть запрещено) и окно ввода кода с простейшими функциями IDE для удобства пользователей.

2.3 Блок результатов

Тестирование производится с помощью автотестов, примеры которых представлены пользователю на странице. В блок результатов выводится информация о результатах данных автотестов. На рисунке 7 представлены результаты выполнения запроса.

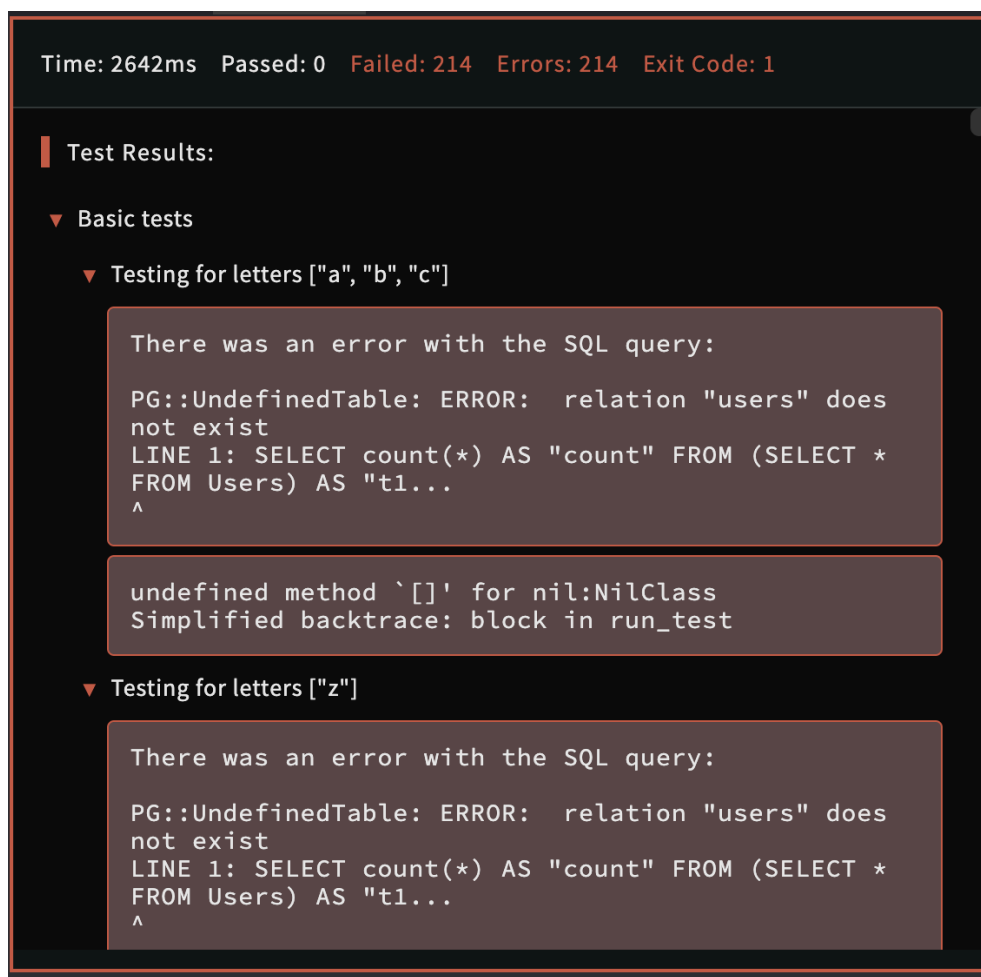


Рисунок 7 – Результаты выполнения запроса

Из данного блока можно получить информацию о времени тестов, количество успешно пройденных, количество проваленных тестов, количество ошибок, результат. Далее представлены ошибки, которые возникали при выполнении скрипта. Данная информация позволяет пользователю сделать выводы о работе написанного скрипта и понять, какие изменения нужно вносить, если при выполнении возникли ошибки.

Для проведения собеседований кандидату имеет смысл показывать только информацию, если скрипт в принципе не выполняется (синтаксические ошибки). Если же скрипт выдает неверный результат и не проходит автотесты, то кандидату об этом знать не нужно. Такая практика является общепринятой и используется, к примеру, на официальных экзаменах Microsoft по MS SQL Server.

2.4 Блок дополнительных функций

Дополнительные функции представлены в виде панели под блоком ввода кода. Блок с данными функциями представлен на рисунке 8.

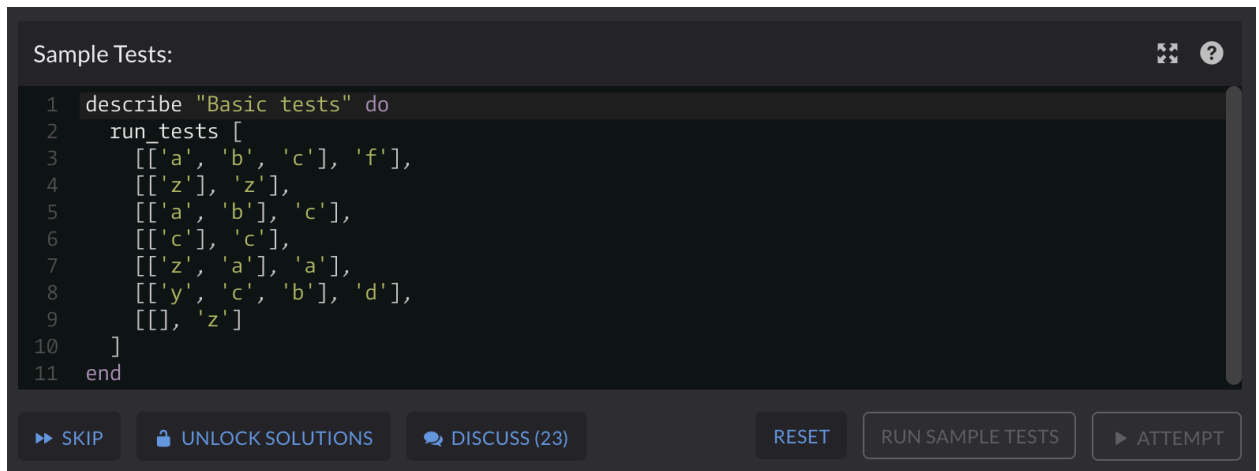


Рисунок 8 – Блок дополнительных функций

В данном блоке присутствует информация о простейших тестах, которыми можно проверить правильность своего решения и функциональные кнопки. С использованием функциональных кнопок можно:

- пропустить задание,
- показать решение,
- обсудить задание,
- перезапустить задание,
- провести простейшие тесты.

Для функционала собеседований необходимыми являются кнопки навигации по заданиям и запуска простейших тестов, то есть тестов синтаксиса и каких-либо других проверок на стороне сервера.

3 Анализ и тестирование функциональности для менеджера

Функциональность менеджера на данном портале представлена страницей создания задания. Данная страница представлена на рисунке 9. Выделим основные функциональные блоки на данной странице:

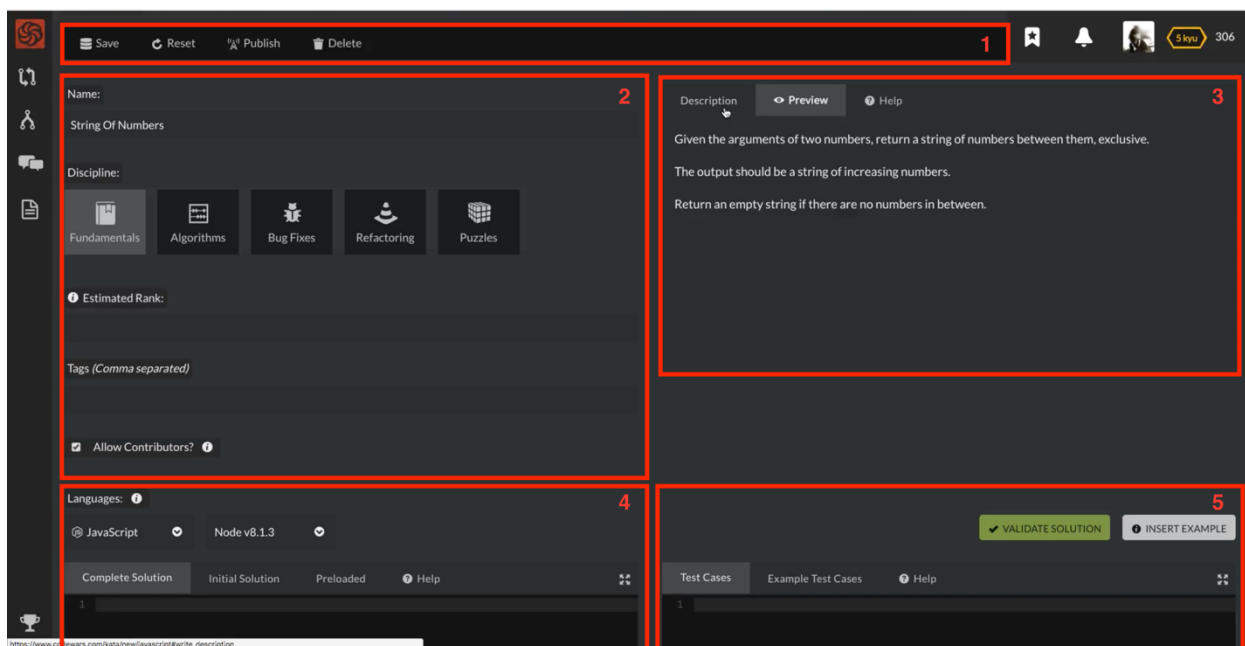


Рисунок 9 – Страница создания задания

- блок управления (1),
- блок информации о задании (2),
- блок задания (3),
- блок решения (4),
- блок тестов (5).

Анализ функциональности страницы создания задания будем проводиться по функциональным блокам, описанным выше.

3.1 Блок управления

Блок управления представлен на рисунке 10.

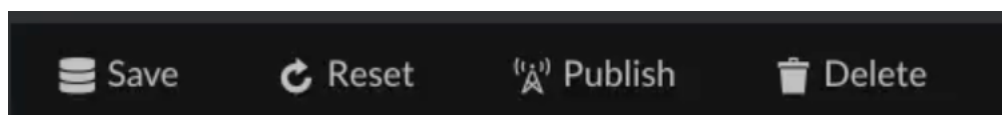


Рисунок 10 – Блок управления

Данный блок содержит 4 функциональных кнопки. Кнопка «Сохранить» - позволяет сохранить задание без его публикации. Кнопка «Сброс» очищает все заполненные поля и можно начать создание задания заново. Кнопка

«Публикация» - осуществляет публикацию задания в открытый доступ.
Кнопка «Удалить» - удаляет задание.

Для функционала собеседований необходимы аналогичные кнопки, так как данные 4 кнопки составляют необходимый функционал для управления заданием.

3.2 Блок информации о задании

Пример данного блока представлен на рисунке 11.

The image shows a dark-themed user interface for creating or editing a task. The form is organized into several sections:

- Name:** A text input field containing "String Of Numbers".
- Discipline:** A section with five selectable options, each represented by an icon and a label: "Fundamentals" (book icon), "Algorithms" (circuit icon), "Bug Fixes" (bug icon), "Refactoring" (funnel icon), and "Puzzles" (cube icon). The "Fundamentals" option is currently selected.
- Estimated Rank:** A section with an information icon and a text input field containing "8 kyu (white)".
- Tags (Comma separated):** A text input field containing "numbers, strings, string, print, count".
- Allow Contributors?** A checkbox that is checked, followed by an information icon.

Рисунок 11 – Блок информации о задании

В данном блоке можно ввести название задание, которое будет отображаться пользователям. Далее представлена возможность выбора дисциплины (типа задания): основы, алгоритмы, исправление багов, рефакторинг кода, пазлы. Затем автору задания предоставляется возможность выбора минимального ранга, с которого доступно задание пользователям (ранг начисляется за верно решенные задания – так данная платформа

ранжирует уровень знаний пользователей). Предпоследнее поле отвечает за теги, которые описывались в пункте 2.1. Завершает данный блок поле разрешения соавторства, которое позволяет разрешить или запретить другим пользователям предлагать изменения в задание.

С точки зрения менеджера ценность имеет только название задания и разрешение соавторства. Остальные же поля не имеют никакой ценности в условиях проведения технического собеседования.

3.3 Блок задания

В блоке задания имеются 3 вкладки: «описание», «предпоказ» и «помощь». Данные вкладки отвечают за редактирование задания, показ задания в виде, в котором оно будет демонстрироваться пользователю и инструкция к работе с этим полем соответственно.

Вкладка «описание» отвечает за полный контроль написания задания, то есть автор может писать в данной вкладке текст, как-либо форматировать его, чтобы получить задание в текстовом виде, которое будет представлено конечному пользователю. На рисунке 12 представлен простейший пример работы с данной вкладкой.

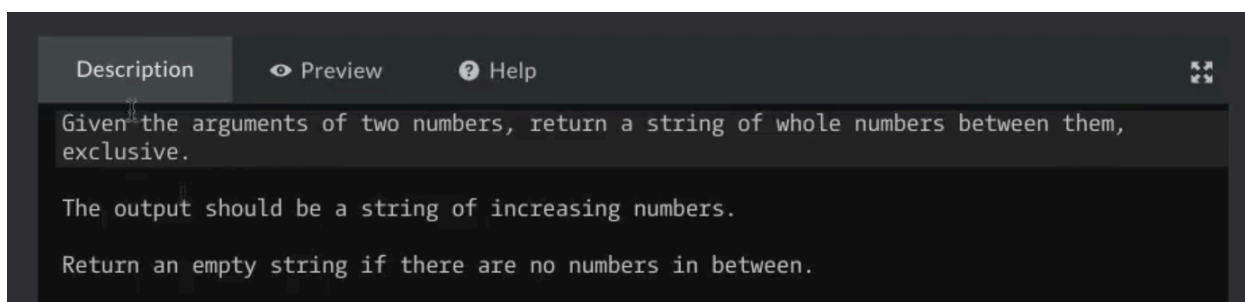


Рисунок 12 – Пример работы с вкладкой «описание»

На вкладке «предпоказ» (рисунок 13) можно получить информацию о том, как будет выглядеть задание, написанное в предыдущей вкладке для конечного пользователя. На данной вкладке представлен текст задания со всеми модификациями (например, выделение текста жирным).

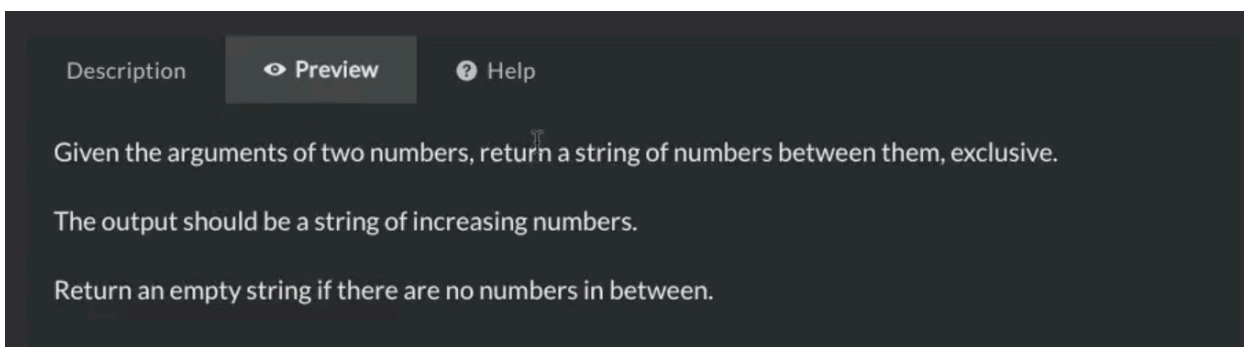


Рисунок 13 – Вкладка «предпоказ»

Вкладка «помощь» (рисунок 14) дает автору возможность получить инструкцию по заполнению данного поля и получить полную и исчерпывающую информацию о возможностях, которые могут быть использованы для формирования текста задания.

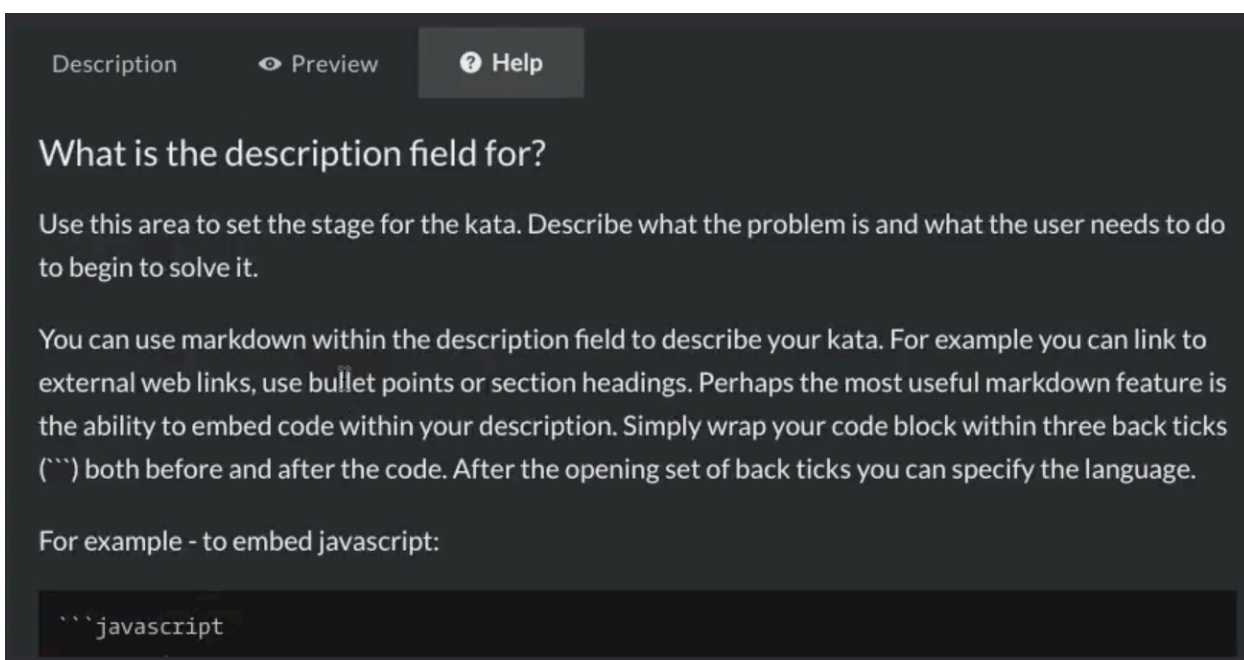


Рисунок 14 – Вкладка «помощь» в блоке задания

Для проведения собеседований все вкладки важны и будут использоваться. Кроме того, необходимо предоставить функцию для добавления картинок в задание. Данная функция имеет ценность именно для заданий по написанию запросов к базе данных, так как с помощью картинку проще всего представить схему объектов базы данных.

3.4 Блок решения

Блок решения представляет из себя выбор языка и версии языка программирования и форма с 4 вкладками для создания решения задания. Данный блок представлен на рисунке 15. Выбор языка и версии языка аналогичен по функциональности выбору для пользователя, представленному на рисунке 6 и был описан ранее. Форма ввода решения состоит из вкладок: «полное решение», «начальное решение», «предзагруженное решение» и «помощь». Данная форма ввода решения так же, как блок решения у пользователя имеет функции простейшей IDE.

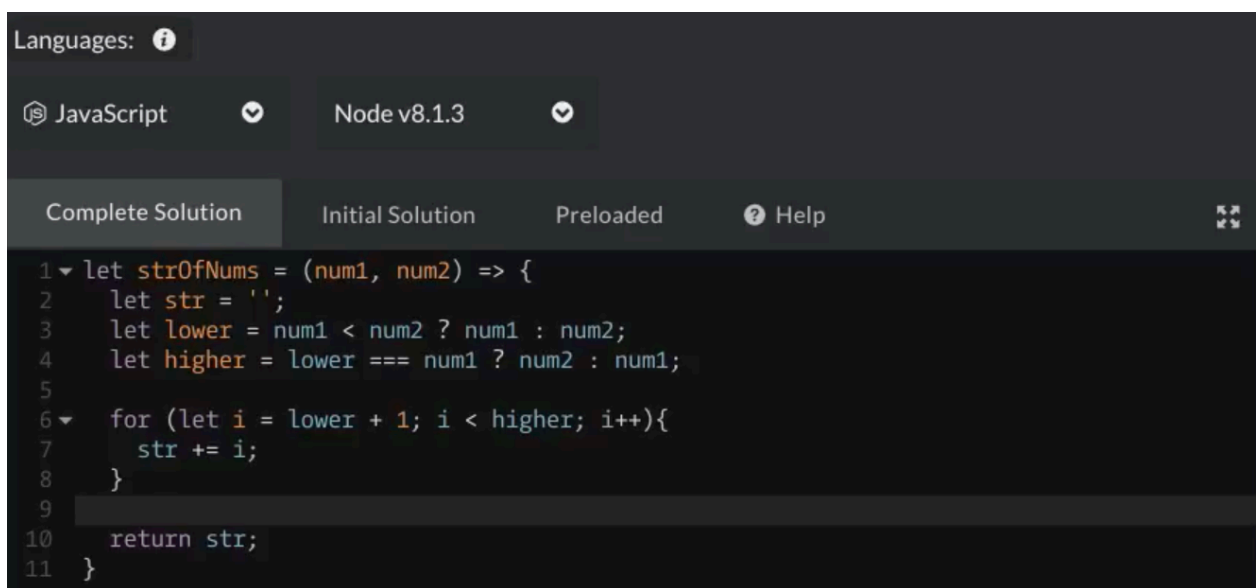
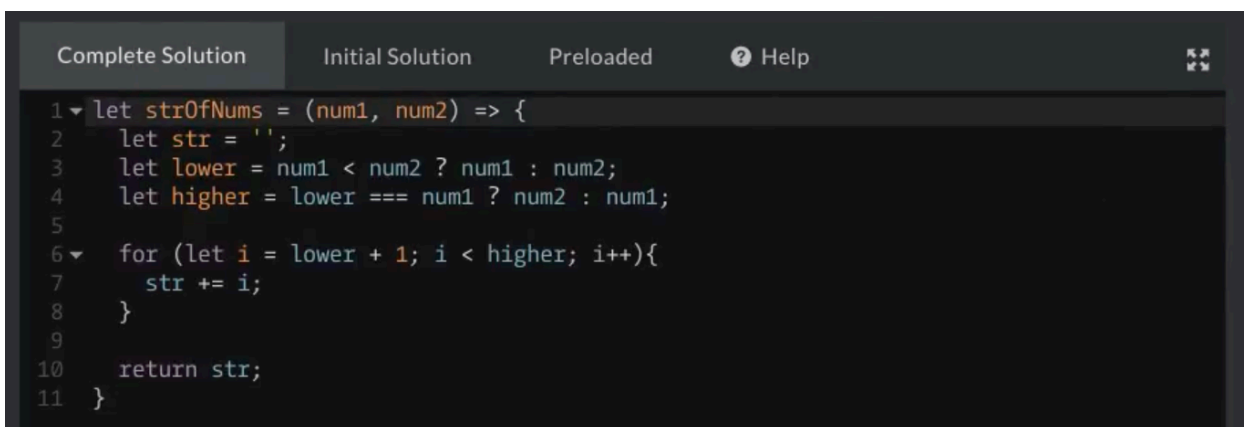


Рисунок 15 – Блок решения

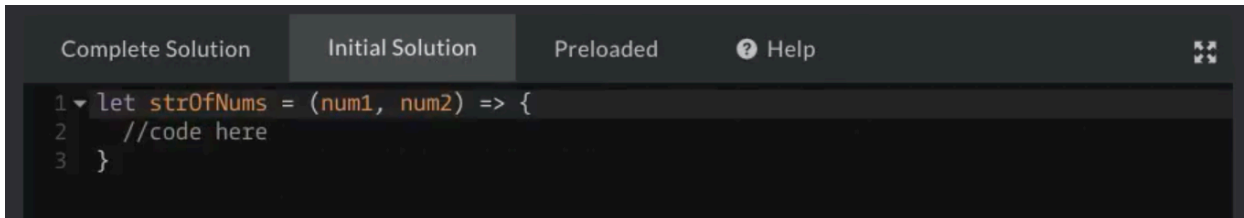
Вкладка «полное решение» (рисунок 16) позволяет ввести эталонное решение задания, написанного прежде. Это решение будет показываться пользователям, если они решат пропустить задание, закончат его решение или нажмут на кнопку показа решения.

The image shows a code editor with four tabs: 'Complete Solution' (selected), 'Initial Solution', 'Preloaded', and '? Help'. The code in the 'Complete Solution' tab is as follows:

```
1 let strOfNums = (num1, num2) => {  
2   let str = '';  
3   let lower = num1 < num2 ? num1 : num2;  
4   let higher = lower === num1 ? num2 : num1;  
5  
6   for (let i = lower + 1; i < higher; i++){  
7     str += i;  
8   }  
9  
10  return str;  
11 }
```

Рисунок 16 – Вкладка «полное решение»

На вкладке «начальное решение» (рисунок 17) автору необходимо написать то, что будет показываться пользователю в блоке написания решения. Это может быть кусок кода, который будет служить подсказкой для решения задания, или комментариев, упрощающий решение задания. При желании и необходимости можно оставить данный блок пустым, тогда у пользователя будет просто пустая первая строка в блоке с решением.

The image shows a code editor with four tabs: 'Complete Solution', 'Initial Solution' (selected), 'Preloaded', and '? Help'. The code in the 'Initial Solution' tab is as follows:

```
1 let strOfNums = (num1, num2) => {  
2   //code here  
3 }
```

Рисунок 17 – Вкладка «начальное решение»

Вкладка «предзагруженное решение» позволяет автору увидеть, как пользователь будет видеть его решение (аналогично с вкладкой «предпоказ» в блоке задания).

Вкладка «помощь» содержит инструкции по работе с данным блоком, для чего он нужен и информацию по правильному написанию эталонных скриптов. Данная вкладка представлена на рисунке 18.

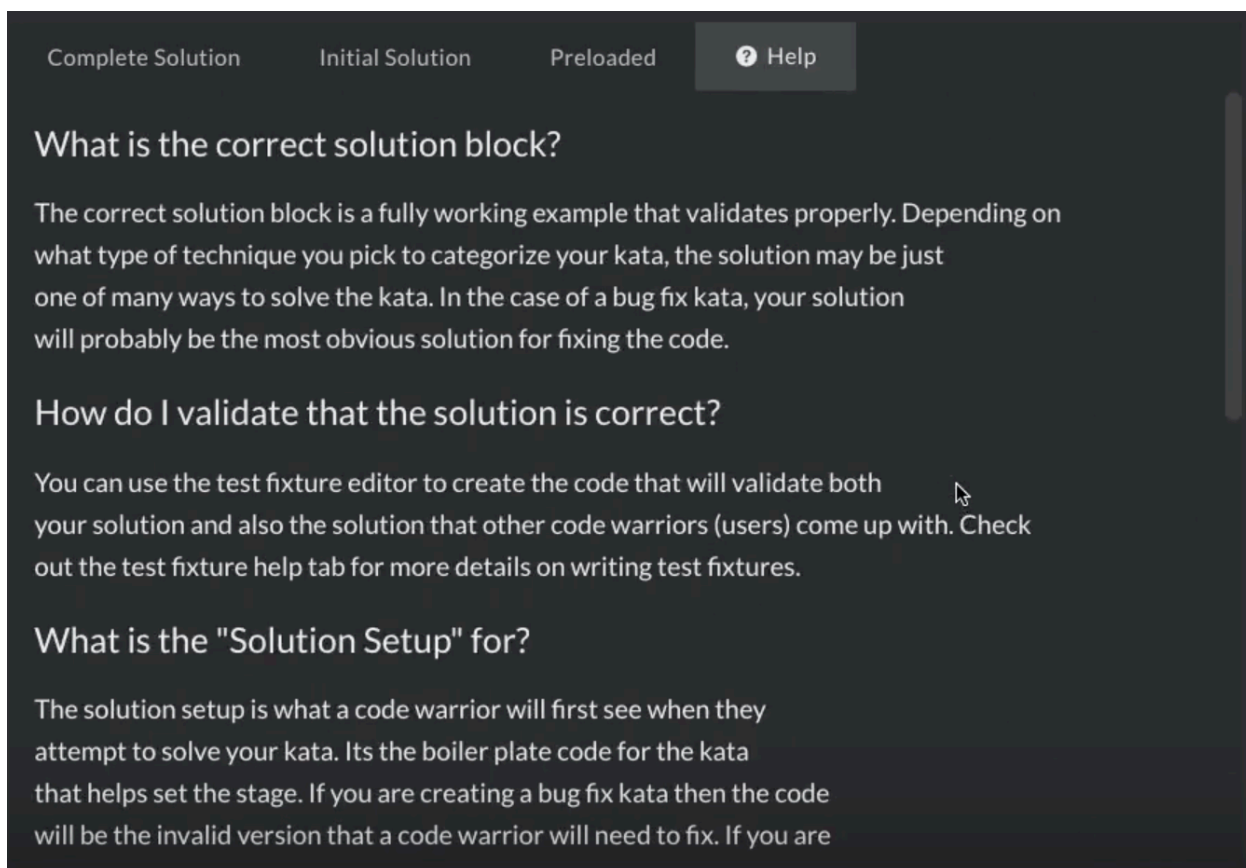


Рисунок 18 – Вкладка «помощь» в блоке решения

Для собеседований являются необходимыми вкладки «полное решение», «начальное решение» и «помощь». Также автору необходима возможность добавлять скрипты для создания объектов базы данных файловом виде или переиспользовать уже созданные объекты.

3.5 Блок тестов

Блок тестов используется для написания функционального тестирования скрипта. В данном блоке автор пишет тесты, которые будут запускаться при нажатии пользователем кнопки для запуска его скрипта. Данный блок состоит из кнопок запуска тестов, помощи, которые запускают тесты и вставляют примеры тестов соответственно, и 3 вкладок: «тесты», «примеры тестов» и «помощь». Данный блок представлен на рисунке 19.

Кнопка запуска тестов запускает все написанные автором тестом, которые указаны во вкладке «тесты». Кнопка помощи покажет примеры кода,

которые могут быть использованы для написания тестов. Функция особенно необходима для авторов, которые пишут тесты впервые.

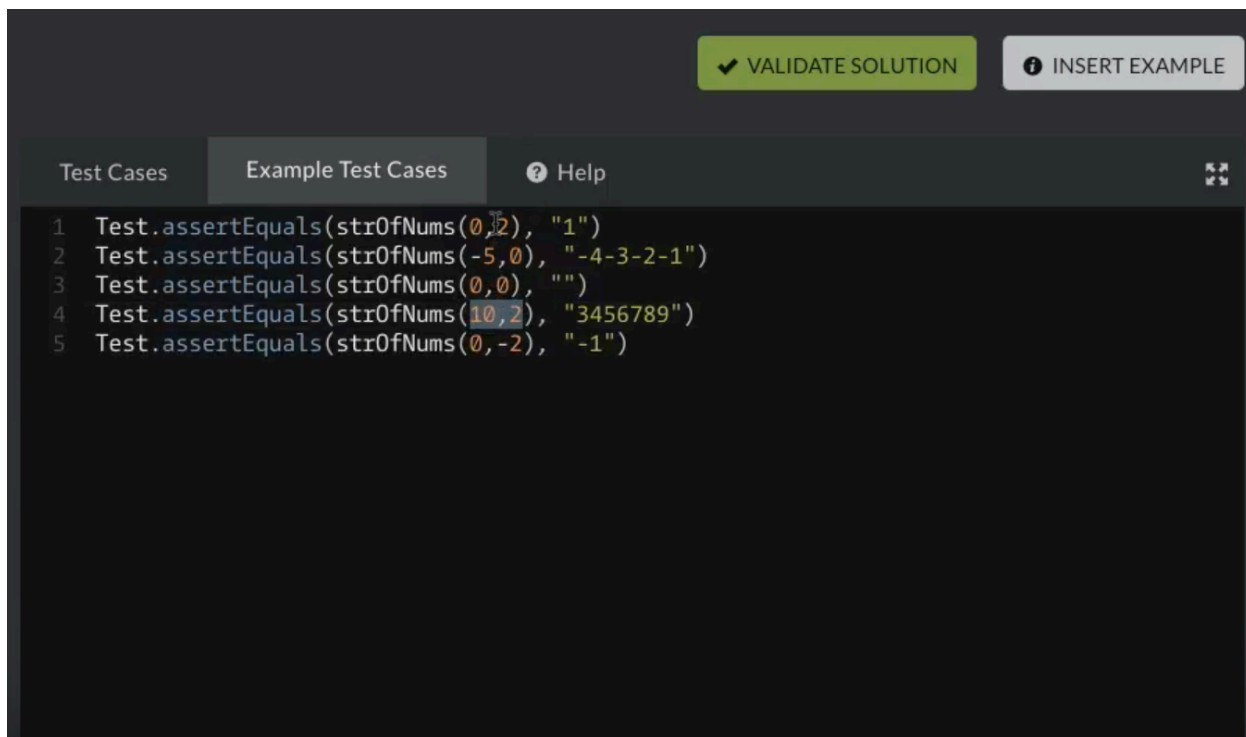
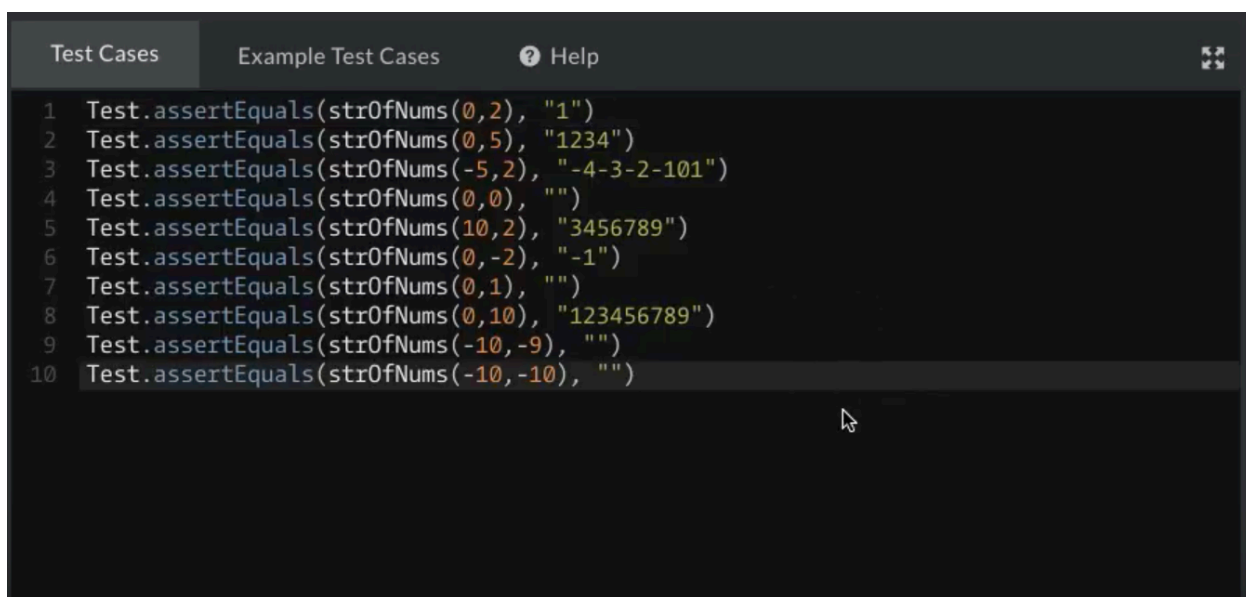


Рисунок 19 – Блок тестов

Вкладка «тесты» (рисунок 20) содержит информацию о тестах, которые должны быть проведены для получения исчерпывающей информации о правильности работы пользовательского скрипта. Скрипт, написанный автором, должен проходить все тесты без ошибок для того, чтобы опубликовать задание. Считается хорошим тоном, если количество тестов приближается к нескольким сотням, содержат «граничные условия» и обработку неверных условий.

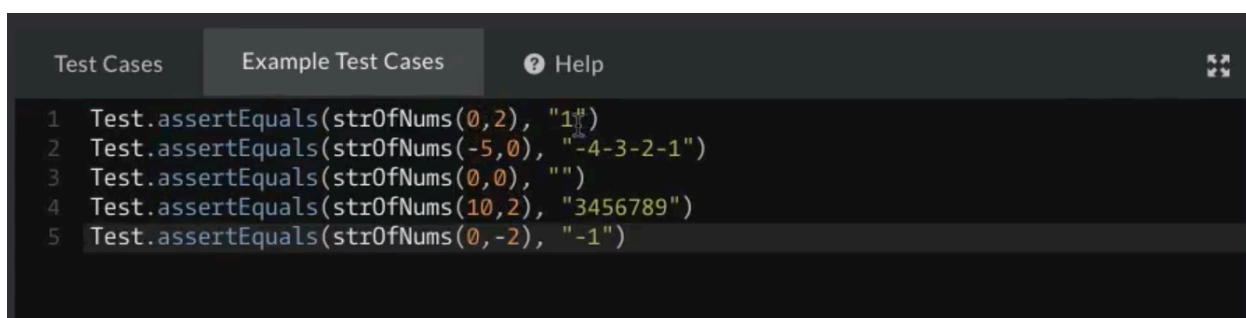
В вкладке «примеры тестов» (рисунок 21) находится информация о тестах, которые будут показаны пользователю в виде примера того, как будет проверяться его решение. Данная вкладка аналогична вкладке «тесты», но содержит лишь малую часть запускаемых тестов.

Вкладка «помощь» содержит инструкцию о работе с остальными вкладками в данном блоке, которая объяснить пользователю цель данного блока, что и как нужно писать в данном блоке и как это должно работать.



```
1 Test.assertEquals(strOfNums(0,2), "1")
2 Test.assertEquals(strOfNums(0,5), "1234")
3 Test.assertEquals(strOfNums(-5,2), "-4-3-2-101")
4 Test.assertEquals(strOfNums(0,0), "")
5 Test.assertEquals(strOfNums(10,2), "3456789")
6 Test.assertEquals(strOfNums(0,-2), "-1")
7 Test.assertEquals(strOfNums(0,1), "")
8 Test.assertEquals(strOfNums(0,10), "123456789")
9 Test.assertEquals(strOfNums(-10,-9), "")
10 Test.assertEquals(strOfNums(-10,-10), "")
```

Рисунок 20 – Вкладка «тесты»



```
1 Test.assertEquals(strOfNums(0,2), "1")
2 Test.assertEquals(strOfNums(-5,0), "-4-3-2-1")
3 Test.assertEquals(strOfNums(0,0), "")
4 Test.assertEquals(strOfNums(10,2), "3456789")
5 Test.assertEquals(strOfNums(0,-2), "-1")
```

Рисунок 21 – Вкладка «пример тестов»

Для собеседований для контроля знаний языка написания запросов данный блок не является необходимым в связи с тем, что результатом работы SQL запроса обычно является таблица, поэтому проверить соответствие таблицы, полученной пользователем к таблице, полученной в результате выполнения запроса автора.

ЗАКЛЮЧЕНИЕ

Анализ систем контроля знаний языка запросов к базам данных, выполненный в данной работе, заключался в анализе систем, представленных на российском и зарубежном рынке. Наиболее подходящей системой оказалась система «codewars», которая в соревновательном формате позволяет изучать различные языки программирования, в том числе и интересующий нас SQL с различными версиями.

Анализ системы был проведен по всем функциональным блокам, присутствующим на портале как с точки зрения пользователя, так и с точки зрения автора заданий. В результате работы можно выделить основные пункты, необходимые для успешного использования системы в ходе технических собеседований.

Для кандидата важно:

- блок задания, который содержит информацию об авторе и непосредственно само условие задания;
- блок ввода решения, содержащий возможность выбора языка, версии языка и форма ввода решения с простейшими функциями IDE;
- блок результатов, содержащий информацию о корректности написанного запроса с точки зрения синтаксиса и возможности запуска введенного запроса;
- блок дополнительных функций для навигации по заданиям и запуска простейших тестов.

Для менеджера важно:

- блок управления, который используется для управления заданием (публикация, сохранение, удаление, очистка);
- блок информации о задании, содержащий всю необходимую информацию о задании (название, разрешение соавторства);

- блок задания, содержащий форму для написания задания, которая содержит несколько вкладок (текст задания, предпросмотр, помощь) для удобства и простоты написания задания;
- блок решения, с информацией об эталонном решении задачи, которое будет использоваться для проверки решений кандидатов, состоящий из вкладок (решение, начальное решение и помощь) и формы для запуска скриптов создания объектов базы данных.

На основании научно-исследовательской работы было разработано техническое задание, расположенное в приложении А.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Основы SQL [Электронный ресурс] <https://proghub.ru/t/sql-basic> (дата обращения 23.11.2019).
2. GeekBrains – образовательный портал [Электронный ресурс] <https://geekbrains.ru/> (дата обращения 23.11.2019).
3. Codewars [Электронный ресурс] <https://www.codewars.com/> (дата обращения 23.11.2019).
4. Codewars.com wiki [Электронный ресурс] <https://www.codewars.com/> (дата обращения 23.11.2019).
5. Teach coding with games: a review of Codewars and CodeCombat [Электронный ресурс] <https://opensource.com/education/15/7/codewars-codecombat-review> (дата обращения 23.11.2019).
6. Why Codewars is the best way to learn a new programming language! [Электронный ресурс] <https://opensource.com/education/15/7/codewars-codecombat-review> (дата обращения 23.11.2019).
7. Code Wars: Creating a Problem / Kata [Электронный ресурс] https://www.youtube.com/watch?v=Ytl4WPJ_zgA (дата обращения 23.11.2019).
8. How to Create a Kata in Codewars [Электронный ресурс] <https://www.youtube.com/watch?v=TZTnP3bZZSU> (дата обращения 23.11.2019).

ПРИЛОЖЕНИЕ А – Техническое задание