VRP Learning Center

Programming

Assignment

Version 1.2

**Test No.1**

# **Для выполнения задания вам помогут**

<https://trailhead.salesforce.com>

<https://resources.docs.salesforce.com/198/latest/en-us/sfdc/pdf/apex_workbook.pdf>

<https://trailhead.salesforce.com/modules/visualforce_fundamentals>

<https://resources.docs.salesforce.com/sfdc/pdf/salesforce_pages_developers_guide.pdf>

<http://www.salesforce.com/us/developer/docs/workbook_database/workbook_database.pdf>

# **Создание орга Разработки**

1. Наберите в браузере

https://developer.salesforce.com/signup

2. Заполните все обязательные поля и нажмите Sign me up.

3. На ваш электронный адрес придет письмо со ссылкой. Перейдите по ссылке и задайте пароль.

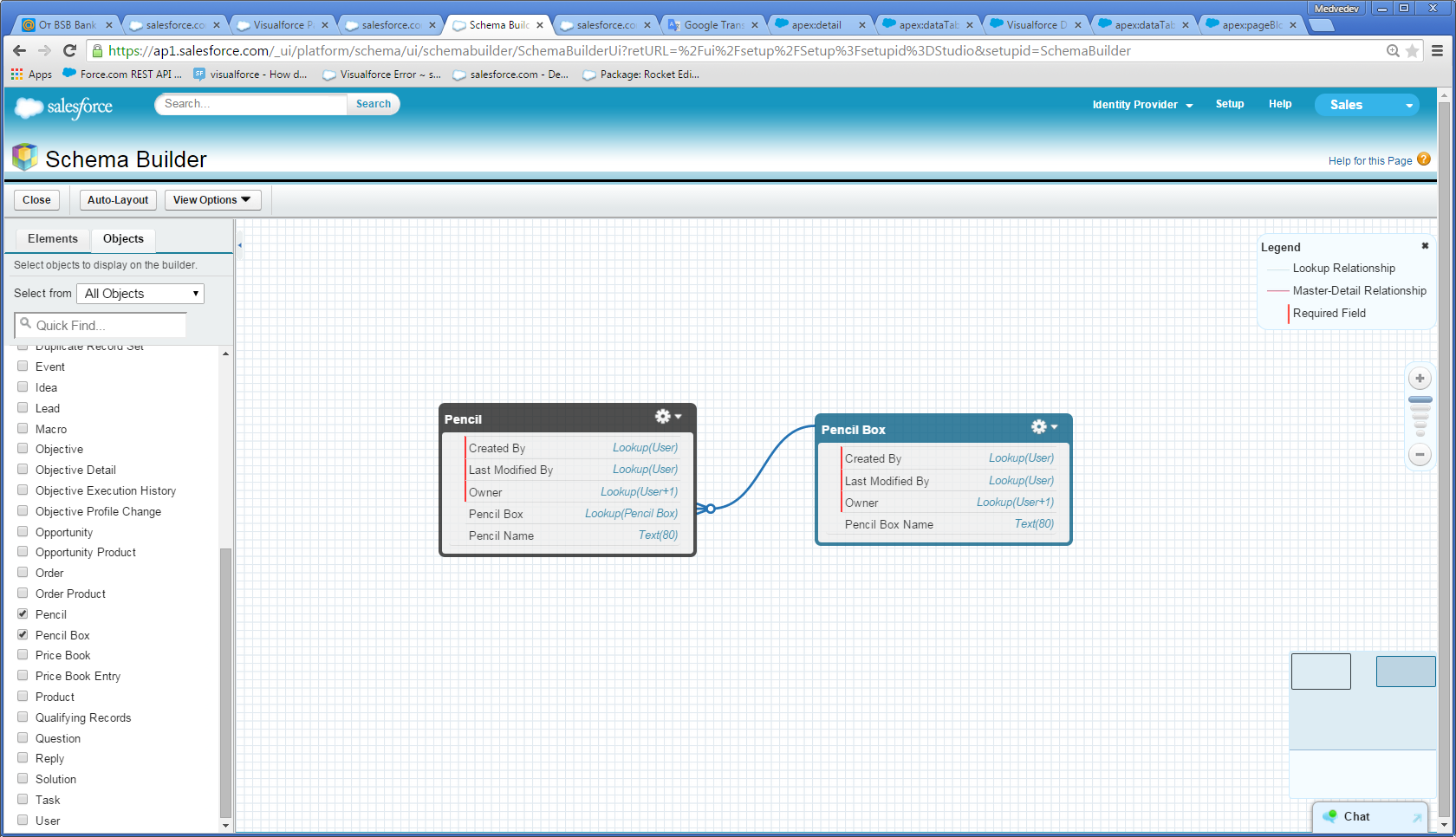
# **Задание:**

## 1. Создать базу данных:

- Создать объект Pencil Box;

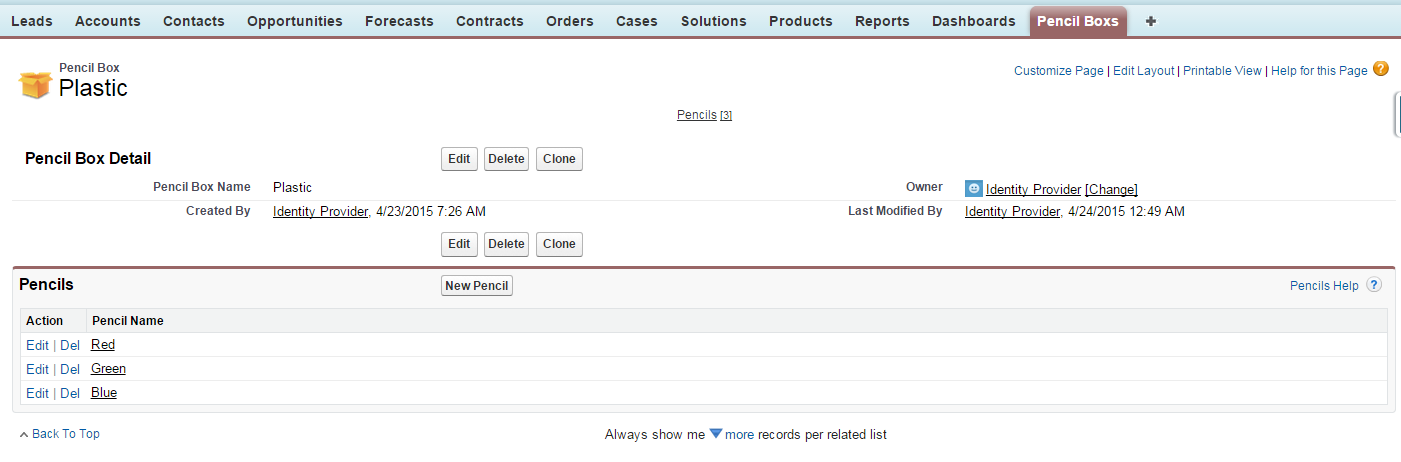
- Создать объект Pencil;

- Создать связь один ко многим между Pencil Box и Pencil;



## 2. Создать Tab для объекта Pencil Box.

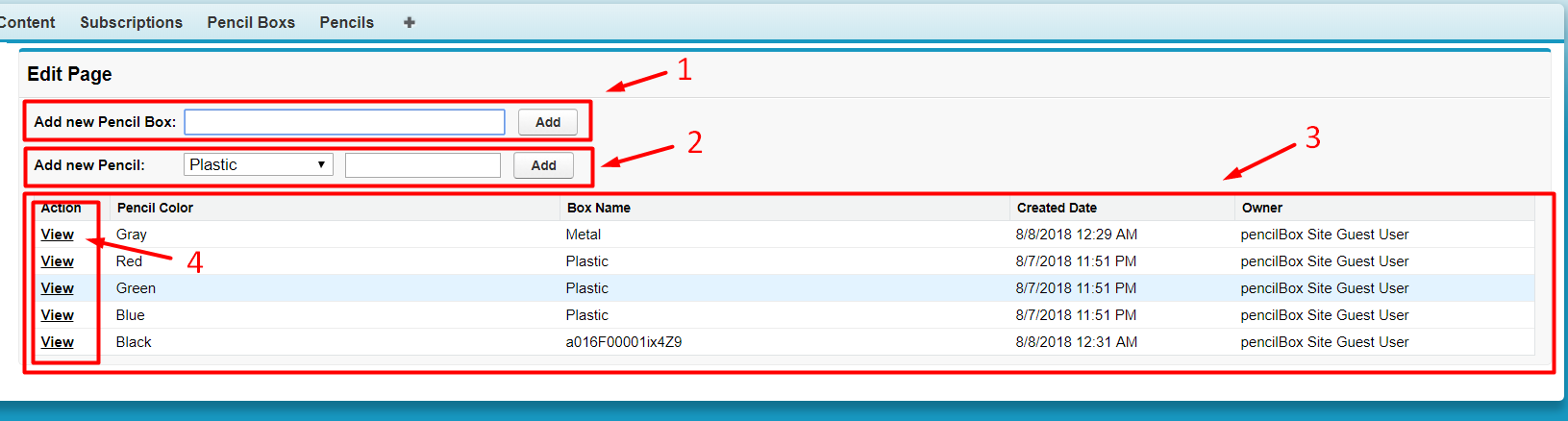
## 3. Создайте запись для объекта Pencil Box и к нему несколько дочерних записей объекта Pencil:



## 4. Создайте Visualforce page и Apex Controller.

Страница должна содержать :

* Поле ввода с именем “Add new Pencil Box” и кнопка Add(1). При нажатии на кнопку должен создаваться новый Pencil Box с именем указанным в поле ввода.
* Поле ввода с именем “Add new Pencil ”, выпадающий список со всеми существующеми Pencil Box и кнопка Add(2). При нажатии на кнопку должен создаваться новый Pencil с именем указанным в поле вода и вложенным в Pencil Box, выбранным в выпадающем списке.
* Таблица в которой отображаются все Pencil записи(3). Каждая запись в таблицу должна содержать ссылкуна страницу с деталями соответствующей Pencil записи(4).



# **Пример работы:**

https://test-task-example-developer-edition.ap4.force.com/pensilbox/

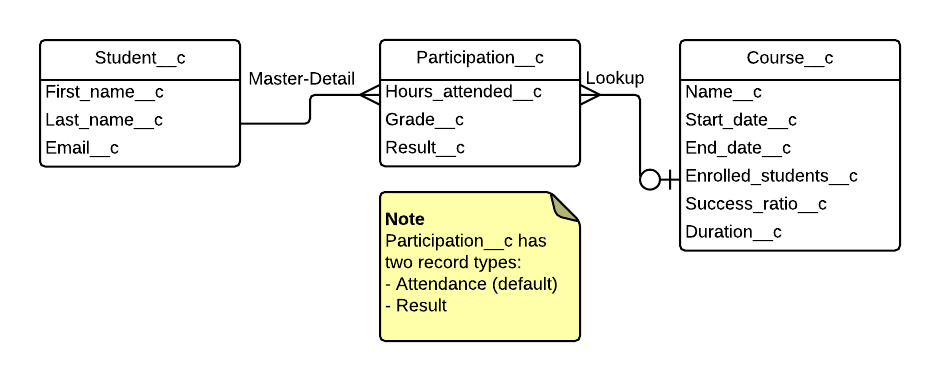
**Test No.2**

# **Scenario**

The Cloud University is a newly created school to educate students around the use of cloud computing technologies. The Cloud University use a Force.com application to track students, courses, attendances and exam results. The app is currently used by professors and other university staff.

Students information is tracked in the Student\_\_c object, which contains basic contact information. Each course held at the university has Course\_\_c record. When a student applies to a course a record is created in the Participation\_\_c table: this object contains one record per student per course, which summarizes the hours attended and, at after the final examination, the result (pass/fail) together with the final grade. A student that get D, E or F, fails the exam.

The full data model is depicted below.



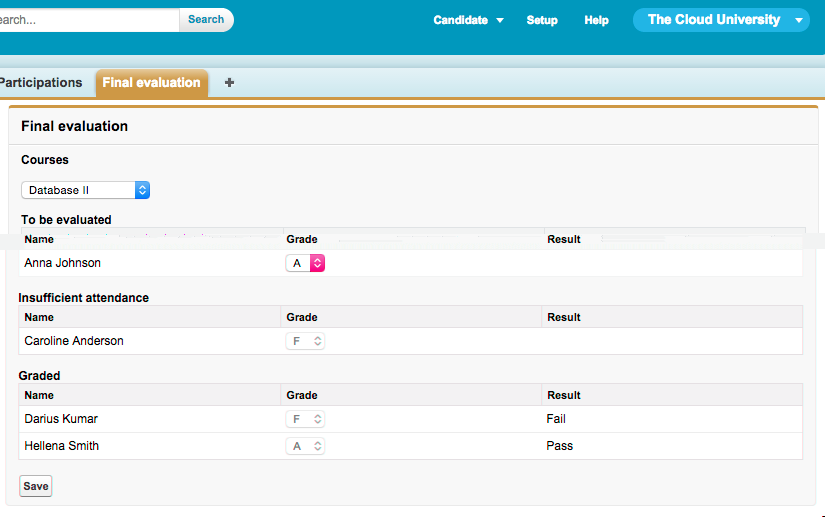
# **Requirements**

## Requirement 1: business logic

In order to enforce data consistency, you are requested to develop a new functionality to update the course summary every time a participation is created or updated. The number of enrolled students (Enrolled\_Students\_\_c) needs to be updated to reflect the overall number of students that are attending the course. The success ratio (Success\_Ratio\_\_c) on the course object (Course\_\_c) needs to be updated to reflect the success rate amongst the students that completed the exam. Success ratio can be calculated as the number of students that passed the course divided by students that completed the course, either with a pass or a fail result. In addition to this, the participation record type should be changed to “Result” when a score is present.

## Requirement 2: student evaluation

Professors asked for a new functionality to make the scoring process easier at the end of each course, so that they can enter all the results at once instead of updating individual records. After a design workshop, the requirement has been outlined in the following wireframe/mockup.



**Please note the following:**

* A picklist displays all the courses that the professor is running
* When a course is selected, the students that are participating in the course are displayed under three sections: “To be evaluated”, “Insufficient attendance” and “Graded”
* Students with at least 50% attendance are showing in the “To be evaluated” section. A picklist allows the professor to select a grade from A to F.
* Students with less than 50% attendance are shown in the “Insufficient attendance” section and their score is defaulted to “F”, with no possibility for the professor to change it.
* Students that have been graded already are shown in the “Graded” section, where the name, the score and the pass/fail is displayed.
* When the Save button is pressed all the grades and the pass/fail are saved for all the students and the page is updated consistently.

## **Evaluation**

In addition to the fulfilment of the requirements, the final evaluation will take into consideration use of native Apex, simplicity and readability of the code, use of programming best practices, robustness of business logic and comprehensiveness of test methods.

## **Guidelines**

* Do not change the data model
* This is meant to be a coding test and therefore is meant to assess your capabilities of resolving issues with code; do not use admin functionality even if in the real world that would be the preferred option (i.e: formula fields, rollup summaries, workflows, etc)
* Keep it simple: don’t over complicate the problem or the solution; you won’t get any extra points for additional features outside of the requirements
* Don’t spend too much on the UI: the solution has to be functional and well coded, doesn’t have to look nice
* Stick to native Visualforce and Apex as much as you can. Do not reinvent the wheel and avoid workarounds
* Stick to best practices
* Write robust unit tests

|  |  |
| --- | --- |
| **For more information:**  Contact your personal assistant Kukenova Galina +375295054450 | **Minsk office**  +375172613422 |