Система генерации отчетных форм

https://bitbucket.org/serg_msuru/sqlreports

15 октября 2016 г.

1 Общее описание

Система предназначена для генерации отчетных форм в реляционной базе данных. Весь пользовательский интерфейс реализуется в виде Web-приложения. Серверная часть должна быть написана на Python с использованием библиотеки Django.

Каждая отчетная форма имеет уникальное имя. Предполагается, что программист, знакомый с языком SQL и структурой базы данных, через Web-интерфейс вводит текст *параметризованного SQL запроса* (то есть SQL запроса, который содержит в некоторых местах имена параметров). Этот запрос используются для выбора данных, необходимых для составления отчета. Конечный пользователь может запросить составление отчета с заданным именем. При поступлении обращения от пользователя на составления отчета происходит следующее.

- Система составляет список параметров, которые используются в SQL-запросе этого отчета.
- Автоматически составляется *форма для ввода значений параметров*. Эта форма показывается пользователю.
- После ввода значений параметров происходит подстановка этих значений в текст запроса и формируется SQL запрос, пригодный для выполнения в конкретной СУБД.
- После получения данных от СУБД формируется *экранная форма отчета*. Предполагается, что отчет представляет собой HTML-страницу, которая содержит таблицу (результат выполнения SQL-запроса). Некоторые ячейки таблицы могут быть оформлены как гиперссылки, переход по которым запускает новый отчет (возможно, с другим именем).
- При необходимости пользователь может сохранить сгенерированный отчет в виде Excel или PDF.

В системе предполагается интерфейс для программиста, интерфейс для конечного пользователя и серверная часть, которая обрабатывает параметризованные запросы.

Далее эти пункты будут описаны подробнее.

2 Описание отчета

Отчет состоит из

- параметризованного SQL запроса;
- описания входных параметров отчета;
- описания выбираемых (выходных) значений;
- шаблонов для генерации PDF или Excel;
- действий по дополнительной обработке отчета.

Описание параметризованных запросов дано в следующем разделе.

2.1 Входные параметры

Входные параметры определяются именем (которое используется в тексте SQL-запроса), типом (строка, число, список значений), названием, более подробным описанием или комментарием, регулярным выражением для проверки корректности значения. Эти данные используются при генерации формы ввода значений. Например, имя «year», название «год», тип — integer, комментарий — «Год принятия сотрудника на работу». Форма ввода данных для выполнения отчета может содержать фразу «Введите значение параметра **год**», а при наведении курсора на название параметра будет всплывать окно с комментарием. Корректность значений может проверяться средствами JavaScript на стороне клиента перед отправкой формы.

Предполагается, что некоторые параметры пользователю вводить разрешено, а некоторые — нет.

2.2 Выбираемые значения

Выбираемые значения описываются именем, названием, действием, которое необходимо совершить при нажатии на соответствующую ячейку отчета (например, ссылка на отчет).

2.3 Шаблоны для генерации PDF

Конечные пользователи любят получать красивые документы. Просмотр отчета в виде HTML-страницы с гиперссылками удобен для специалиста, но руководителю принято показывать хорошо оформленные документы. Должна быть возможность для генерации PDF версий отчета. Одно из

решений — связать с отчетом latex-файл. Желательно, чтобы начальная версия такого latex шаблона генерироваль автоматически, чтобы избавить администратора от рутинной работы.

Этот вопрос требует уточнения.

3 Язык параметризованных SQL-запросов

Должна поддерживаться подстановка конкретных значений вместо имени параметра.

```
SELECT employee_id
FROM employee
WHERE employee_name = '%(full_name)s'
```

Один запрос может включать несколько параметров, а параметр с одим именем может несколько раз встречаться в одном запросе. В последнем случае все вхождения параметра должны заменяться на одно и то же значение. Например,

```
SELECT employee_id
FROM employee
WHERE employee_hired = %(year)s
      OR
      (employee_hired < %(year)s AND employee_age < 30)
```

Должна быть возможность задания параметров, которые изменяют запрос. Например, значением параметра является имя выбираемого поля:

```
SELECT %(!field)s AS category, %(!fun)s(salary) AS value
FROM employee
GROUP BY %(!field)s
```

Имена таких параметров должны начинаться с восклицательного знака (опасность!). В качестве значения параметра `!field` можно поставить имя поля, а в качестве значения `!fun` — имя агрегирующей функции, например MAX или AVG.

3.1 Включения одного запроса в другой

Должна быть возможность вставить текст одного параметризованного запроса в другой. Подстановка должна выполняться в момент выполнения запроса, то есть в момент составления отчета.

Если включаемый параметризованный запрос содержит параметры, то команда включения должна допускать подстановку значений параметров. Например,

```
SELECT subq.fullname, count(*)
FROM (
/*+ include list_by_name(param1=param3,
```

```
      param2=14, param3='abc') +*/
) subq
GROUP BY subq.fullname
```

В этом примере предполагается, что параметризованный запрос с именем `list_by_name` имеет три входных параметра, которые называются param1, param2 и param3. При включении производится подстановка констант вместо параметров param2 и param3, а значение для param1 берется из входного параметра param3 основного запроса. При выполнении такого отчета должна появится форма для ввода значений параметров с указанием только param3.

3.2 Условные операторы

```
SELECT emp.department_id, sum(employee_weight)
FROM employee emp
/*+ if top_floors=1 +*/
     , department dep
WHERE emp.department_id = dep.department_id
      AND
      dep.department_floor > 10
/*+ endif +*/
GROUP BY emp.department_id
HAVING sum(emp.employee_weight) > 1000
```

```
/*+ if ... +*/
/*+ include ... +*/
/*+ else +*/
...
/*+ endif +*/
```

4 Интерфейс конечного пользователя

Конечный пользователь при работе с системой использует:

- форму для ввода параметров отчета, которая появляется при выполнении отчета;
- форму просмотра результатов выполнения отчета (HTML-версия отчетной формы);
- PDF-версию.

5 Интерфейс для программиста

Должна быть возможность

- ввести параметризованный SQL запрос (запрос, который используется в других запросах и отчетах);

- полностью описать отчет (в соответствии с разделом 2);
- просмотреть список запросов, которые используются (через механизм включения запросов) для составления данного отчета;
- просмотреть список отчетов, которые вызываются из данного (при нажатии на ячейки таблицы);
- просмотреть список запросов, которые включают данный параметризованный запрос.