|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий (ИТ)

Кафедра Инструментального и прикладного программного обеспечения

(ИиППО)

**ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №11**

**по дисциплине**

«Технологии обработки транзакций клиент-серверных приложений»

Выполнил студент группы ИКБО-20-21 Хитров Н.С.

ПринялМаличенко С.В.

Москва 2024**ЗАДАНИЕ НА ПРАКТИЧЕСКУЮ РАБОТУ**

1. Функция empapi.run, написанная на языке Python, принимает параметр типа jsonb. Добавьте трансформацию, чтобы избежать преобразований в текстовый вид и обратно.
2. Отправляйте пользователю, совершившему покупку, письмо-подтверждение с указанием суммы. Для отправки пользуйтесь уже готовой функцией public.sendmail (посмотрите ее определение) или напишите свою. Посылать письмо внутри транзакции покупки неправильно: транзакция может быть оборвана по какой-либо причине, а письмо уже уйдет. Воспользуйтесь механизмом фоновых заданий: в транзакции добавляйте задание на отправку письма. В таком случае оно будет отправлено, только если транзакция завершится успешно.

**РЕФЕРАТ**

Отчёт 13 страниц, 5 рисунков, 4 листингов, 5 источников.

POSTGRESQL, ДОВЕРЕННЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ, ТРАНЗАКЦИИ, СЕРВЕРНОЕ ПРОГРАММИРОВАНИЕ, SQL

Объектом разработки является база данных.

Цель работы – изучение и применение концепции транзакций в контексте баз данных с использованием языков программирования.

В процессе работы производилось исследование концепции транзакций в контексте баз данных с использованием языков программирования.

Результатом являются сведения о концепции транзакций в контексте баз данных с использованием языков программирования.

**ВВЕДЕНИЕ**

[ВВЕДЕНИЕ 6](#_Toc165774090)

[ВЫПОЛНЕНИЕ ПРАКТИЧЕСКОЙ РАБОТЫ 7](#_Toc165774091)

[ЗАКЛЮЧЕНИЕ 12](#_Toc165774094)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 13](#_Toc165774095)

**ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ**

БД – база данных;

СУБД – система управления базами данных.

**ВВЕДЕНИЕ**

В современном мире, где данные играют важнейшую роль в различных сферах, от бизнеса до повседневной жизни, вопрос вариативности написания кода на стороне сервера является одним из ключевых и влияет на возможности взаимодействия с системой и хранимыми данными. Для этого в PostgreSQL используются различные языки программирования, используемые для написания функций, процедур, триггеров и т.д.

Цель данной практической работы состоит в изучении концепции транзакций в контексте баз данных и применении их на практике с использованием языков программирования.

**ВЫПОЛНЕНИЕ ПРАКТИЧЕСКОЙ РАБОТ****Ы**

* 1. **Выполнение задания 1**

При выполнении 1 задания функция empapi.run была изменена согласно листингу 1. Изменения позволяют автоматически конвертировать text в формат jsonb.

Листинг 1 - Код функции myrun

CREATE OR REPLACE FUNCTION myrun(func text, params jsonb)

RETURNS text

TRANSFORM FOR TYPE jsonb

AS $python$

p = plpy.prepare("SELECT \* FROM " + plpy.quote\_ident(func) + "($1)", ["jsonb"])

r = p.execute([params])

cols = r.colnames()

collen = {col: len(col) for col in cols}

for i in range(len(r)):

for col in cols:

if len( str(r[i][col]) ) > collen[col]:

collen[col] = len( str(r[i][col]) )

res = ""

res += " ".join( [col.center(collen[col]," ") for col in cols]) + "\n"

res += " ".join( ["-"\*collen[col] for col in cols]) + "\n"

for i in range(len(r)):

res += " ".join( [str(r[i][col]).ljust(collen[col]," ") for col in cols]) + "\n"

return res

$python$ LANGUAGE plpython3u VOLATILE;

Код вспомогательной функции для тестирования, выводящей данные таблицы представлен в листинге 2.

Листинг 2 - Код функции helper\_func

CREATE OR REPLACE FUNCTION helper\_func(params jsonb)

RETURNS TABLE(id INT, value TEXT)

AS $$

DECLARE

\_table text;

\_query text;

BEGIN

\_table := params ->> 'table';

\_query := 'SELECT id, value::text FROM ' || quote\_ident(\_table);

RETURN QUERY EXECUTE \_query;

END;

$$ LANGUAGE plpgsql;

Результат работы функций (Рисунок 1).

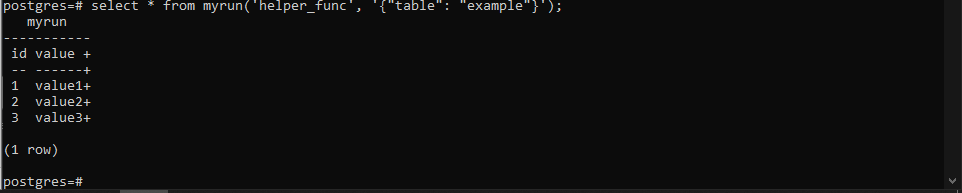


Рисунок 1 - Демонстрация работы функции myrun

* 1. **Выполнение задания 2**

Для выполнения данного задания использовалась реализация функции sendmail, представленная в листинге 3, которая позволяла пересылать сообщения с использованием smtp google.

При выполнении задания 2 была реализована функция sendmail, выполняющая пересылку сообщений по протоколу smtp (Листинг3).

Листинг 3 - Код функции sendmail

CREATE OR REPLACE FUNCTION public.sendmail(from\_addr text, to\_addr text, subj text, msg text)

RETURNS void

LANGUAGE plpython3u

AS $function$

import smtplib

server = smtplib.SMTP('smtp.gmail.com', 587)

server.starttls()

server.login('bevalerry@gmail.com', 'srchfgkoaquvqguh')

server.sendmail(

from\_addr,

to\_addr,

"\r\n".join([

"From: %s" % from\_addr,

"To: %s" % to\_addr,

"Content-Type: text/plain; charset=\"UTF-8\"",

"Subject: %s" % subj,

"\r\n%s" % msg

]).encode('utf-8')

)

server.quit()

$function$;

На листинге 4 представлена функция checkout выполняющая формирование данных для отправки на почту и помещение соответствующего задания в очередь.

Листинг 4 - Код функции checkout

CREATE OR REPLACE FUNCTION public.checkout(user\_id bigint)

RETURNS void

AS $$

DECLARE

params jsonb;

BEGIN

SELECT jsonb\_build\_object(

'from\_addr', 'bevalerry@gmail.com',

'to\_addr', customer.email,

'subj', 'Order Confirmation for User ' || customer.name,

'msg', format(

E'Hello %s!\nThank you for your order.\nYour order total is $%s.',

customer.name,

total

)

)

INTO params

FROM customer

JOIN (

SELECT sales\_order.customer\_id, SUM(item.total) AS total

FROM sales\_order

JOIN item ON sales\_order.order\_id = item.order\_id

WHERE sales\_order.customer\_id = user\_id

GROUP BY sales\_order.customer\_id

) AS order\_totals ON order\_totals.customer\_id = customer.customer\_id;

PERFORM public.run\_program(program\_id => 1, params => params);

END;

$$ LANGUAGE plpgsql VOLATILE;

На рисунках 2-3 представлено помещение задания в очередь и само задание в очереди.

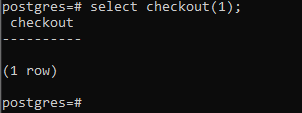


Рисунок 2 - Вызов функции checkout

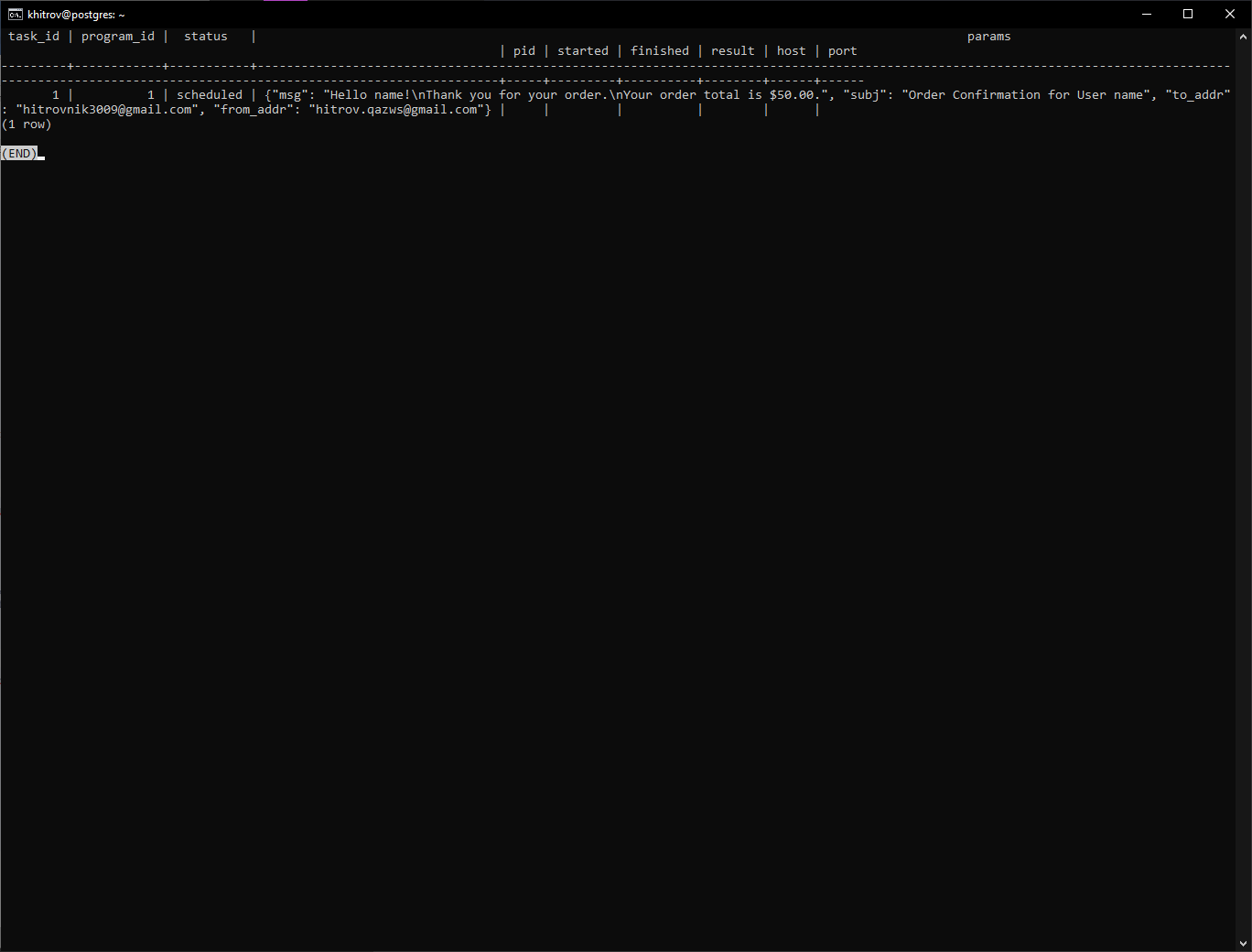


Рисунок 3 - Содержимое таблицы tasks

Далее для выполнения отправки сообщений из очереди на почту былы использована функция execute\_tasks (Рисунок 4).

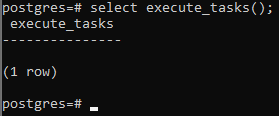


Рисунок 4 - Вызов функции execute\_tasks

Полученное сообщение представлено на рисунке 5.

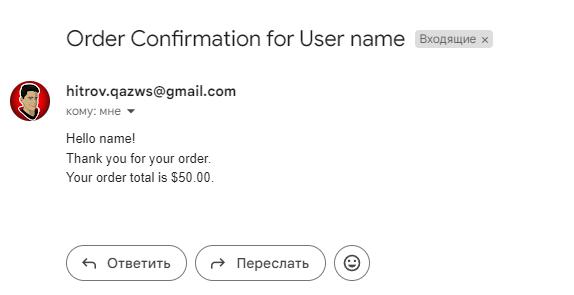


Рисунок 5 - Полученное сообщение

# ЗАКЛЮЧЕНИЕ

В ходе выполнения практической работы были получены практические навыки по применению различных языков серверного программирования для решения различных задач на стороне сервера.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. PostgreSQL: Официальная документация [Электронный ресурс] – URL: https://www.postgresql.org/docs/ (дата обращения: 14.02.2024).
2. PostgreSQL Tutorial for Beginners [Электронный ресурс] – URL: https://www.tutorialspoint.com/postgresql/index.htm (дата обращения: 14.02.2024).
3. PostgreSQL: Википедия [Электронный ресурс] – URL: https://en.wikipedia.org/wiki/PostgreSQL (дата обращения: 14.02.2024).
4. Лузанов П.В. и др. Postgres. Первое знакомство. [Электронный ресурс] – URL: https://postgrespro.ru/education/books/introbook (дата обращения: 22.02.2024)
5. Новиков Б. А. Лекции Основы технологий баз данных. [Электронный ресурс] – URL: https://postgrespro.ru/education/university/dbtech (дата обращения: 22.02.2024)