МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тихоокеанский государственный университет»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Разработка системы логирования и бекапа

Лабораторная работа №4

по дисциплине «Проектирование приложений баз данных»

Выполнил студент Пшеничный Д. О.

Факультет, группа ФКФН, ПО(аб)-81

Руководитель работы  **Федосеев А.А.**

Хабаровск – 2021г.

**Постановка задачи**

Для БД, разработанной в практической работе № 1, реализовать систему логирования действий пользователя по изменению данных. Действия пользователя сохраняются в таблице вида: R(*LogUsers*)={User (Имя пользователя), TimeAction (дата и время действия), TableAction (изменяемая таблица), Action (оператор SQL по изменению данных)}. Студенты с чётными номерами в журнале реализуют логирование, используя механизм хранимых функций, а с нечётными — триггеров. Продемонстрировать работу системы логирования путём выполнения различных действий по изменению данных, используя консоль PostgreSQL.

Система бэкапа выполняется с использованием утилиты *pg\_dump* [1]. Для реализации процедуры бэкапа разработать приложение, осуществляющее создание резервной копии базы данных без участия пользователя. Время первого запуска процедуры создания резервной копии и промежуток времени задаются через ключи командной строки.

Имя файла резервной копии формируется по синтаксису: “<имя БД>\_<дата>\_<время>” Продемонстрировать работу создания резервных копий и восстановления БД на их основе.

**Четный вариант.** Создание системы логирования с использованием хранимых функций.

**Текст хранимых функций**

--Вставка в Course

CREATE OR REPLACE FUNCTION insertIntoCourse(INT)

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'course', 'INSERT');

INSERT INTO Course (number\_course) VALUES ($1);

END;

$$ LANGUAGE plpgsql;

--Вставка в Discipline

CREATE OR REPLACE FUNCTION insertIntoDiscipline(VARCHAR(50))

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'discipline', 'INSERT');

INSERT INTO Discipline (name\_disc) VALUES ($1);

END;

$$ LANGUAGE plpgsql;

--Вставка в Speciality

CREATE OR REPLACE FUNCTION insertIntoSpeciality(VARCHAR(50))

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'speciality', 'INSERT');

INSERT INTO Speciality (name\_spec) VALUES ($1);

END;

$$ LANGUAGE plpgsql;

--Вставка в Uch\_Plan

CREATE OR REPLACE FUNCTION insertIntoUchPlan(VARCHAR(50), INT)

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'uch\_plan', 'INSERT');

INSERT INTO Uch\_Plan (name\_uch, Speciality\_id\_spec) VALUES ($1, $2);

END;

$$ LANGUAGE plpgsql;

--Вставка в Gruppa

CREATE OR REPLACE FUNCTION insertIntoGruppa(VARCHAR(50), DATE, INT, INT)

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'gruppa', 'INSERT');

INSERT INTO Gruppa (name\_group, enter\_date, Uch\_Plan\_id\_uch, Speciality\_id\_spec) VALUES ($1, $2, $3, $4);

END;

$$ LANGUAGE plpgsql;

--Вставка в Gupr\_elem

CREATE OR REPLACE FUNCTION insertIntoGupr\_elem(VARCHAR(50))

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'gupr\_elem', 'INSERT');

INSERT INTO Gupr\_elem (name\_gupr\_elem) VALUES ($1);

END;

$$ LANGUAGE plpgsql;

--Вставка в Gupr

CREATE OR REPLACE FUNCTION insertIntoGupr(INT, INT)

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'gupr', 'INSERT');

INSERT INTO Gupr (duration, Gupr\_elem\_id\_gupr\_elem) VALUES ($1, $2);

END;

$$ LANGUAGE plpgsql;

--Вставка в Kafedra

CREATE OR REPLACE FUNCTION insertIntoKafedra(VARCHAR(50))

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'kafedra', 'INSERT');

INSERT INTO Kafedra (name\_kaf) VALUES ($1);

END;

$$ LANGUAGE plpgsql;

--Вставка в Kaf\_Rasp

CREATE OR REPLACE FUNCTION insertIntoKaf\_Rasp(INT, INT, INT)

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'kaf\_rasp', 'INSERT');

INSERT INTO Kaf\_Rasp (Speciality\_id\_spec, Discipline\_id\_disc, Kafedra\_id\_kaf) VALUES ($1, $2, $3);

END;

$$ LANGUAGE plpgsql;

--Вставка в Semestr

-- CREATE OR REPLACE FUNCTION insertIntoSemestr()

-- RETURNS VOID AS $$

-- BEGIN

-- INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

-- VALUES (current\_user, now(), 'semestr', 'INSERT');

-- INSERT INTO Semestr () VALUES ();

-- END;

-- $$ LANGUAGE plpgsql;

--Вставка в Uch\_Load

CREATE OR REPLACE FUNCTION insertIntoUch\_Load(INT, INT, INT, INT, INT)

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'uch\_load', 'INSERT');

INSERT INTO Uch\_Load (hours, Uch\_Plan\_id\_uch, Speciality\_id\_spec, Semestr\_id\_semestr, Discipline\_id\_disc) VALUES ($1, $2, $3, $4, $5);

END;

$$ LANGUAGE plpgsql;

--Вставка в number\_weeks

CREATE OR REPLACE FUNCTION insertIntoNumber\_Weeks(INT)

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'number\_weeks', 'INSERT');

INSERT INTO number\_weeks (number) VALUES ($1);

END;

$$ LANGUAGE plpgsql;

--Удаление из Course

CREATE OR REPLACE FUNCTION deleteFromCourse(INT)

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'course', 'DELETE');

DELETE FROM Course

WHERE number\_course=$1;

END;

$$ LANGUAGE plpgsql;

--Удаление из Discipline

CREATE OR REPLACE FUNCTION deleteFromDiscipline(VARCHAR(50))

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'discipline', 'DELETE');

DELETE FROM Discipline

WHERE name\_disc=$1;

END;

$$ LANGUAGE plpgsql;

--Удаление из Speciality

CREATE OR REPLACE FUNCTION deleteFromSpeciality(VARCHAR(50))

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'speciality', 'DELETE');

DELETE FROM Speciality

WHERE name\_spec=$1;

END;

$$ LANGUAGE plpgsql;

--Удаление из Uch\_Plan

CREATE OR REPLACE FUNCTION deleteFromUchPlan(VARCHAR(50), INT)

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'uch\_plan', 'DELETE');

DELETE FROM Uch\_Plan

WHERE name\_uch=$1 AND Speciality\_id\_spec=$2;

END;

$$ LANGUAGE plpgsql;

--Удаление из Gruppa

CREATE OR REPLACE FUNCTION deleteFromGruppa(VARCHAR(50), DATE, INT, INT)

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'gruppa', 'DELETE');

DELETE FROM Gruppa

WHERE name\_group=$1 AND enter\_date=$2 AND Uch\_Plan\_id\_uch=$3 AND Speciality\_id\_spec=$4;

END;

$$ LANGUAGE plpgsql;

--Удаление из Gupr\_elem

CREATE OR REPLACE FUNCTION deleteFromGupr\_elem(VARCHAR(50))

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'gupr\_elem', 'DELETE');

DELETE FROM Gupr\_elem

WHERE name\_gupr\_elem=$1;

END;

$$ LANGUAGE plpgsql;

--Удаление из Gupr

CREATE OR REPLACE FUNCTION deleteFromGupr(INT, INT)

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'gupr', 'DELETE');

DELETE FROM Gupr

WHERE duration=$1 AND Gupr\_elem\_id\_gupr\_elem=$2;

END;

$$ LANGUAGE plpgsql;

--Удаление из Kafedra

CREATE OR REPLACE FUNCTION deleteFromKafedra(VARCHAR(50))

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'kafedra', 'DELETE');

DELETE FROM Kafedra

WHERE name\_kaf=$1;

END;

$$ LANGUAGE plpgsql;

--Удаление из Kaf\_Rasp

CREATE OR REPLACE FUNCTION deleteFromKaf\_Rasp(INT, INT, INT)

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'kaf\_rasp', 'DELETE');

DELETE FROM Kaf\_Rasp

WHERE Speciality\_id\_spec=$1 AND Discipline\_id\_disc=$2 AND Kafedra\_id\_kaf=$3;

END;

$$ LANGUAGE plpgsql;

--Удаление из Semestr

-- CREATE OR REPLACE FUNCTION deleteFromSemestr()

-- RETURNS VOID AS $$

-- BEGIN

-- INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

-- VALUES (current\_user, now(), 'semestr', 'DELETE');

-- INSERT INTO Semestr () VALUES ();

-- END;

-- $$ LANGUAGE plpgsql;

--Удаление из Uch\_Load

CREATE OR REPLACE FUNCTION deleteFromUch\_Load(INT, INT, INT, INT, INT)

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'uch\_load', 'DELETE');

DELETE FROM Uch\_Load

WHERE hours=$1 AND Uch\_Plan\_id\_uch=$2 AND Speciality\_id\_spec=$3 AND Semestr\_id\_semestr=$4 AND Discipline\_id\_disc=$5;

END;

$$ LANGUAGE plpgsql;

--Удаление из number\_weeks

CREATE OR REPLACE FUNCTION deleteFromNumber\_Weeks(INT)

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'number\_weeks', 'DELETE');

DELETE FROM number\_weeks

WHERE number=$1;

END;

$$ LANGUAGE plpgsql;

--Изменение Course

CREATE OR REPLACE FUNCTION updateCourse(INT, INT)

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'course', 'UPDATE');

UPDATE Course

SET number\_course=$2

WHERE number\_course=$1;

END;

$$ LANGUAGE plpgsql;

--Изменение Discipline

CREATE OR REPLACE FUNCTION updateDiscipline(VARCHAR(50), VARCHAR(50))

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'discipline', 'UPDATE');

UPDATE Discipline

SET name\_disc=$2

WHERE name\_disc=$1;

END;

$$ LANGUAGE plpgsql;

--Изменение Speciality

CREATE OR REPLACE FUNCTION updateSpeciality(VARCHAR(50), VARCHAR(50))

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'speciality', 'UPDATE');

UPDATE Speciality

SET name\_spec=$2

WHERE name\_spec=$1;

END;

$$ LANGUAGE plpgsql;

--Изменение Uch\_Planupdate

CREATE OR REPLACE FUNCTION updateUchPlan(VARCHAR(50), VARCHAR(50), INT, INT)

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'uch\_plan', 'UPDATE');

UPDATE Uch\_Plan

SET name\_uch=$2, Speciality\_id\_spec=$4

WHERE name\_uch=$1 AND Speciality\_id\_spec=$3;

END;

$$ LANGUAGE plpgsql;

--Изменение Gruppa

CREATE OR REPLACE FUNCTION updateGruppa(VARCHAR(50), VARCHAR(50), DATE, DATE, INT, INT, INT, INT)

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'gruppa', 'UPDATE');

UPDATE Gruppa

SET name\_group=$2, enter\_date=$4, Uch\_Plan\_id\_uch=$6, Speciality\_id\_spec=$8

WHERE name\_group=$1 AND enter\_date=$3 AND Uch\_Plan\_id\_uch=$5 AND Speciality\_id\_spec=$7;

END;

$$ LANGUAGE plpgsql;

--Изменение Gupr\_elem

CREATE OR REPLACE FUNCTION updateGupr\_elem(VARCHAR(50), VARCHAR(50))

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'gupr\_elem', 'UPDATE');

UPDATE Gupr\_elem

SET name\_gupr\_elem=$2

WHERE name\_gupr\_elem=$1;

END;

$$ LANGUAGE plpgsql;

--Изменение Gupr

CREATE OR REPLACE FUNCTION updateGupr(INT, INT, INT, INT)

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'gupr', 'UPDATE');

UPDATE Gupr

SET duration=$2, Gupr\_elem\_id\_gupr\_elem=$4

WHERE duration=$1 AND Gupr\_elem\_id\_gupr\_elem=$3;

END;

$$ LANGUAGE plpgsql;

--Изменение Kafedra

CREATE OR REPLACE FUNCTION updateKafedra(VARCHAR(50), VARCHAR(50))

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'kafedra', 'UPDATE');

UPDATE Kafedra

SET name\_kaf=$2

WHERE name\_kaf=$1;

END;

$$ LANGUAGE plpgsql;

--Изменение Kaf\_Rasp

CREATE OR REPLACE FUNCTION updateKaf\_Rasp(INT, INT, INT, INT, INT, INT)

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'kaf\_rasp', 'UPDATE');

UPDATE Kaf\_Rasp

SET Speciality\_id\_spec=$2, Discipline\_id\_disc=$4, Kafedra\_id\_kaf=$6

WHERE Speciality\_id\_spec=$1 AND Discipline\_id\_disc=$3 AND Kafedra\_id\_kaf=$5;

END;

$$ LANGUAGE plpgsql;

--Изменение Semestr

-- CREATE OR REPLACE FUNCTION updateSemestr()

-- RETURNS VOID AS $$

-- BEGIN

-- INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

-- VALUES (current\_user, now(), 'semestr', 'UPDATE');

-- INSERT INTO Semestr () VALUES ();

-- END;

-- $$ LANGUAGE plpgsql;

--Изменение Uch\_Load

CREATE OR REPLACE FUNCTION updateUch\_Load(INT, INT, INT, INT, INT, INT, INT, INT, INT, INT)

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'uch\_load', 'UPDATE');

UPDATE Uch\_Load

SET hours=$2, Uch\_Plan\_id\_uch=$4, Speciality\_id\_spec=$6, Semestr\_id\_semestr=$8, Discipline\_id\_disc=$10

WHERE hours=$1 AND Uch\_Plan\_id\_uch=$3 AND Speciality\_id\_spec=$5 AND Semestr\_id\_semestr=$7 AND Discipline\_id\_disc=$9;

END;

$$ LANGUAGE plpgsql;

--Изменение number\_weeks

CREATE OR REPLACE FUNCTION updateNumber\_Weeks(INT, INT)

RETURNS VOID AS $$

BEGIN

INSERT INTO Log\_Users(User\_Name, TimeAction, TableAction, Action)

VALUES (current\_user, now(), 'number\_weeks', 'UPDATE');

UPDATE number\_weeks

SET number=$2

WHERE number=$1;

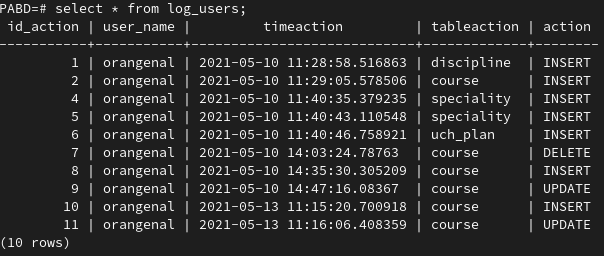
END;

$$ LANGUAGE plpgsql;

**Команда для восстановления базы из бекапа**

[orangenal@matebook-fedora bin]$ ./pg\_restore /home/orangenal/dumper/dumper/PABD\_2021-05-13\_11\:06\:02 -c -d PABD

**Скриншот лог таблицы**



**Код дампера**

import psutil

import os

import sys

import datetime

message = ""

location = sys.argv[0][:-7]

def dump():

filename = location + sys.argv[1] + "\_" + datetime.datetime.now().strftime('%Y-%m-%d\_%H:%M:%S')

os.system(location + "/dump.sh " + filename + " " + sys.argv[1] + " " + location)

return "Dump created with name " + filename

for proc in psutil.process\_iter():

name = proc.name()

if name == "postgres":

message = dump()

break

if message == "":

message = "Running postgres is not found!"

f = open(location + "dump\_log", "a")

f.write(message + " " + location + "\n")

f.close()