ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| старший преподаватель |  |  |  |  |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| ХРАНИМЫЕ ПРОЦЕДУРЫ |
| по курсу: ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4132 |  |  |  | Е.И. Куконен |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2023

**Цель**: Создание процедур.

**Задание**:

создать в БД хранимые процедуры, реализующие:

— вставку с пополнением справочников (получаем ссылку на внешний ключ по значению данных из родительской таблицы, если данных нет- добавляем в родительскую, затем вставляем в дочернюю);

— удаление с очисткой справочников – удаление данных из родительской таблицы, если после удаления данных из дочерней у строки родительской больше нет зависимых (удаляется информация о студенте, если в его группе нет больше студентов, запись удаляется из таблицы с перечнем групп);

— каскадное удаление (удаление всех зависимых данных);

— вычисление и возврат значения агрегатной функции (т.к. агрегатная функция дает единственный результат) (задача- вернуть данные из процедуры/функции);

— формирование статистики во временной таблице. (задача- работа с временными таблицами).

Содержание отчета:

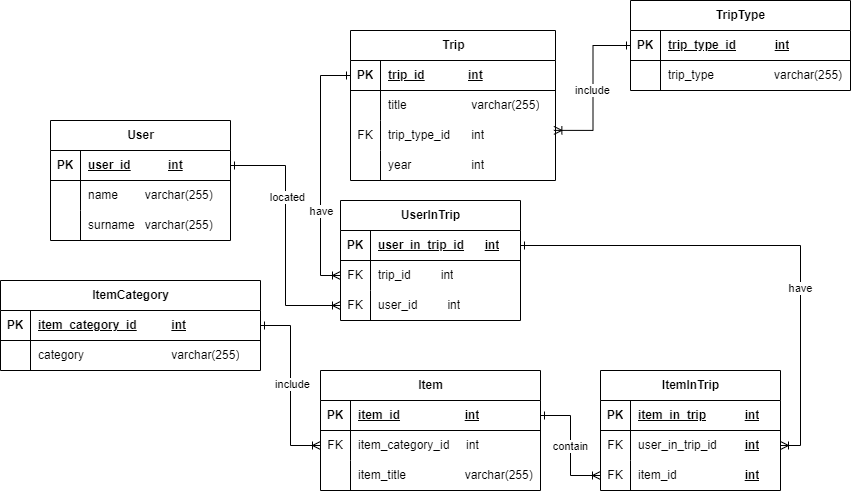
— физическую модель БД.

—назначение, тексты ХП и их вызовов;l

— наборы данных, возвращаемые ХП (для вставки и удаления скриншоты до и после).

**Ход работы**:

Диаграмма:



1. Вставка с пополнением справочников

Назначение:

Вставка в таблицу UserInTrip с заполнением User, Trip, TripType

Код:

CREATE OR REPLACE PROCEDURE insert\_into\_userintrip(

user\_name VARCHAR(255),

user\_surname VARCHAR(255),

trip\_title VARCHAR(255),

trip\_year INT,

trip\_type VARCHAR(255)

)

LANGUAGE plpgsql

AS $$

DECLARE

user\_id\_new INT;

trip\_type\_id\_new INT;

trip\_id\_new INT;

BEGIN

-- user

IF exists(SELECT \* FROM Users WHERE name = user\_name AND surname = user\_surname) THEN

SELECT id INTO user\_id\_new

FROM Users

WHERE name = user\_name AND surname = user\_surname;

ELSE BEGIN

INSERT INTO Users (name, surname)

VALUES (user\_name, user\_surname)

RETURNING id INTO user\_id\_new;

END;

END IF;

-- triptype

IF exists(SELECT \* FROM TripType WHERE type = trip\_type) THEN

SELECT id INTO trip\_type\_id\_new

FROM TripType

WHERE type = trip\_type;

ELSE BEGIN

INSERT INTO TripType (type)

VALUES (trip\_type)

RETURNING id INTO trip\_type\_id\_new;

END;

END IF;

-- trip

IF exists(SELECT \* FROM Trip WHERE title = trip\_title AND year = trip\_year AND type\_id = trip\_type\_id\_new) THEN

SELECT id INTO trip\_id\_new

FROM Trip

WHERE title = trip\_title AND year = trip\_year AND type\_id = trip\_type\_id\_new;

ELSE BEGIN

INSERT INTO Trip (title, year, type\_id)

VALUES (trip\_title, trip\_year, trip\_type\_id\_new)

RETURNING id INTO trip\_id\_new;

END;

END IF;

-- userintrip

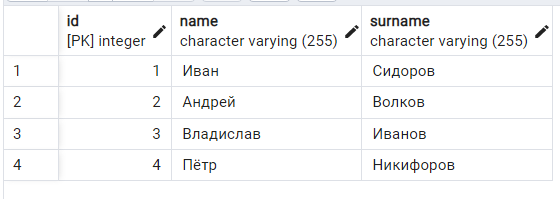
INSERT INTO UserInTrip (user\_id, trip\_id)

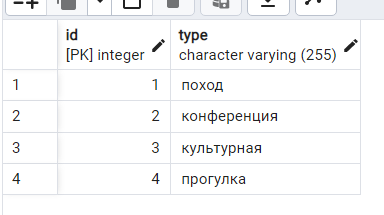
VALUES (user\_id\_new, trip\_id\_new);

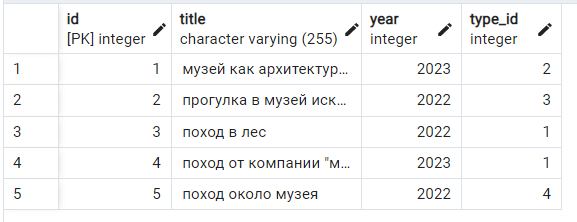
END;

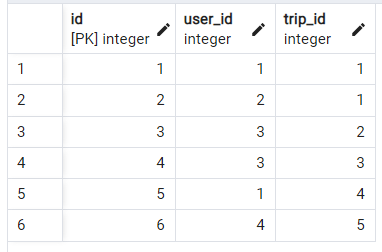
$$;

call insert\_into\_userintrip('Пётр', 'Никифоров', 'поход около музея', 2022, 'прогулка')









1. Удаление с очисткой справочников

Назначение:

Удаление из Trip с очисткой TripType

Код:

CREATE OR REPLACE PROCEDURE clear\_trip\_with\_triptype(

trip\_id INT

)

LANGUAGE plpgsql

AS $$

DECLARE

trip\_type\_id INT;

BEGIN

SELECT type\_id INTO trip\_type\_id FROM Trip WHERE id = trip\_id;

DELETE FROM Trip WHERE id = trip\_id;

IF NOT EXISTS (SELECT \* FROM Trip WHERE type\_id = trip\_type\_id) THEN

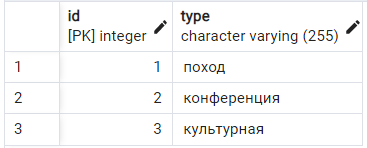
DELETE FROM TripType WHERE id = trip\_type\_id;

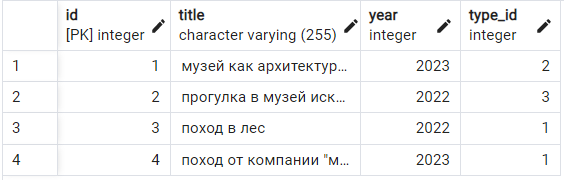
END IF;

END;

$$;

call clear\_trip\_with\_triptype(5);





1. Каскадное удаление.

Назначение:

Удаление из User с очисткой UserInTrip, ItemInTrip

Код:

CREATE OR REPLACE PROCEDURE cascade\_user(

cascade\_user\_id INT

)

LANGUAGE plpgsql

AS $$

DECLARE

trip\_type\_id INT;

BEGIN

DELETE FROM ItemInTrip

WHERE user\_in\_trip\_id IN (

SELECT id FROM UserInTrip uit WHERE user\_id = cascade\_user\_id

);

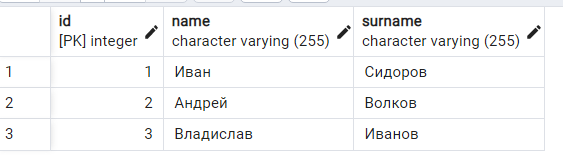
DELETE FROM UserInTrip WHERE user\_id = cascade\_user\_id;

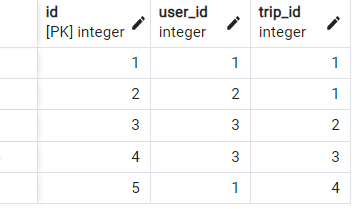
DELETE FROM Users WHERE id = cascade\_user\_id;

END;

$$;

call cascade\_user(4);





1. Вычисление и возврат значения агрегатной функции

Назначение:

Вычисление количества пользователей

Код:

CREATE OR REPLACE FUNCTION users\_count() RETURNS INTEGER AS $$

DECLARE

user\_count INTEGER;

BEGIN

SELECT COUNT(\*) INTO user\_count FROM Users;

RETURN user\_count;

END;

$$ LANGUAGE plpgsql;

SELECT users\_count();

Как процедура:

CREATE OR REPLACE PROCEDURE users\_count\_like\_procedure(

out users\_count INTEGER

) AS $$

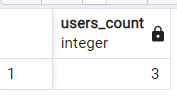
BEGIN

SELECT COUNT(\*) INTO users\_count FROM Users;

END;

$$ LANGUAGE plpgsql;

call users\_count\_like\_procedure(NULL);



1. Создание таблицы со статистикой

Назначение:

Создание временной таблицы с процентом участия пользователя в поездках

Код:

DROP FUNCTION generate\_user\_statistic;

CREATE OR REPLACE FUNCTION generate\_user\_statistic()

RETURNS TABLE (

id INT,

username VARCHAR(255),

trip\_count INT,

trip\_percent NUMERIC

) AS $$

DECLARE

all\_trip\_count INT;

BEGIN

CREATE TEMPORARY TABLE IF NOT EXISTS UserStatisctic (

user\_id INT,

user\_name VARCHAR(255),

trip\_count INT DEFAULT 0,

trip\_percent NUMERIC DEFAULT 0

);

-- заполнение строк id и ФИ пользователя

INSERT INTO UserStatisctic (user\_id, user\_name)

SELECT u.id, u.name || ' ' || u.surname FROM Users u;

-- Добавление количества

UPDATE UserStatisctic

SET trip\_count = us.trip\_count

FROM (

SELECT user\_id, COUNT(trip.id) AS trip\_count FROM UserInTrip

INNER JOIN Trip ON UserInTrip.trip\_id = Trip.id

GROUP BY user\_id

) AS us

WHERE UserStatisctic.user\_id = us.user\_id;

-- Получение количества поездок

SELECT SUM(us.trip\_count) INTO all\_trip\_count FROM UserStatisctic us;

-- Заполнение статистикой

UPDATE UserStatisctic

SET trip\_percent = COALESCE((UserStatisctic.trip\_count::NUMERIC / NULLIF(all\_trip\_count, 0) \* 100), 0)::NUMERIC(4,2);

-- Возврат таблицы из функции

RETURN QUERY SELECT \* FROM UserStatisctic;

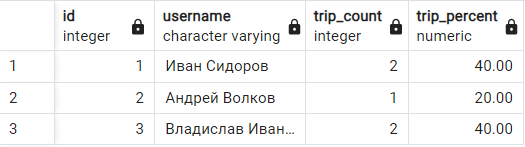
-- Удаление временной таблицы

DROP TABLE IF EXISTS UserStatisctic;

END;

$$ LANGUAGE plpgsql;

SELECT \* FROM generate\_user\_statistic();



**Вывод**: были созданы процедуры.