

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего
образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе №5 «Процедуры, функции, триггеры в PostgreSQL»

по дисциплине «Проектирование и реализация баз данных»

Автор: Захарчук А.И.

Факультет: ИКТ

Группа: К3241

Преподаватель: Говорова М.М.



Санкт-Петербург 2023

Оглавление

Вариант 1. БД «Отель»	3
Ход работы:	4
Вывод.....	12

Цель работы: овладеть практическими создания и использования процедур, функций и триггеров в базе данных PostgreSQL.

Оборудование: компьютерный класс.

Программное обеспечение: СУБД PostgreSQL, pgadmin 4.

Практическое задание:

1. Создать процедуры/функции согласно индивидуальному заданию (часть 4).

2.1. Модифицировать триггер (триггерную функцию) на проверку корректности входа и выхода сотрудника (см. Практическое задание 1 Лабораторного практикума (Приложение)) с максимальным учетом «узких» мест некорректных данных по входу и выходу).

2.2. Создать авторский триггер по варианту индивидуального задания.

Вариант 1. БД «Отель»

Описание предметной области: Отели сети находятся в разных городах. Цены на номера одного типа во всех отелях одинаковы и зависят от типа номера и количества мест. Номер может быть забронирован, занят или свободен. При заезде в отель постояльцы проходят регистрацию. Информация о регистрации постояльцев отеля (выехавших из отеля) хранится в течение года и 1 января удаляется в архив.

Номера ежедневно убираются горничными, для чего составляется график уборки номеров. Ежедневно каждому номеру присваивается статус “убран”, “не убран”.

Цены на номера могут меняться.

БД должна содержать следующий минимальный набор сведений: Адрес отеля. Название отеля. Номер комнаты. Тип комнаты. Количество мест. Удобства. Цена комнаты за сутки проживания. Имя постояльца. Фамилия постояльца. Отчество постояльца. Адрес постоянного проживания. Дата заезда. Дата отъезда. Ежедневный график уборки номеров горничными.

Горничные нанимаются в отели сети сезонно или постоянно. Необходимо хранить информацию по договору найма: номер договора, дата заключения, дата окончания действия, если договор срочный, условия.

Дополнить исходные данные информацией: по бронированию комнаты; по сотруднику, который регистрирует постояльца в отеле в день заезда; по оплате проживания; по составу удобств в комнате; по акциям, доступным при бронировании (скидки).

Задание 4. Создайте хранимые процедуры:

- для увеличения цены всех номеров на 5 %, если в отеле нет свободных номеров;
- для получения информации о свободных одноместных номерах отеля на завтрашний день;
- бронирования двухместного номера в гостинице на заданную дату и количество дней проживания.

Задание 5. Создайте необходимые триггеры.

Ход работы:

1. Процедура для увеличения цены всех номеров на 5 %, если в отеле нет свободных номеров

```
drop procedure if exists raise_prices_by_five_percent;
```

```
CREATE PROCEDURE raise_prices_by_five_percent(hotel_code_param INTEGER)
```

```
LANGUAGE plpgsql
```

```
AS $$
```

```
BEGIN
```

```
    IF (SELECT COUNT(*) FROM hotel WHERE code = hotel_code_param) > 0
```

```
    THEN
```

```
        UPDATE price
```

```
        SET base_price = base_price * 1.05
```

```
        WHERE room_type_code IN
```

```
        (
```

```
            SELECT DISTINCT r.room_type_code
```

```
            FROM room r
```

```
            JOIN hotel h
```

```
                ON r.hotel_code = h.code
```

```
            WHERE h.code = hotel_code_param
```

```
        );
```

```
    END IF;
```

```
END;
```

```
$$
```

```
call raise_prices_by_five_percent(3);
```

Query		Query History				
1	SELECT *					
2	FROM price					
3	WHERE room_type_code IN					
4	(
5	SELECT DISTINCT r.room_type_code					
6	FROM room r					
7	JOIN hotel h					
8	ON r.hotel_code = h.code					
9	WHERE h.code = 3					
10);					
11						
Data Output		Messages				
	id [PK] bigint	room_type_code bigint	base_price numeric	starts_at timestamp without time zone	ends_at timestamp without time zone	
1	6	6	42000	2023-10-04 20:02:43	2024-02-05 00:48:35	
2	9	9	45000	2023-10-21 03:57:43	2023-12-28 13:24:27	
3	12	12	36000	2023-10-12 14:38:33	2023-10-22 16:31:25	
4	14	14	37000	2023-10-19 06:41:48	2024-05-16 02:14:46	

Query		Query History				
1	SELECT *					
2	FROM price					
3	WHERE room_type_code IN					
4	(
5	SELECT DISTINCT r.room_type_code					
6	FROM room r					
7	JOIN hotel h					
8	ON r.hotel_code = h.code					
9	WHERE h.code = 3					
10);					
Data Output		Messages				
	id [PK] bigint	room_type_code bigint	base_price numeric	starts_at timestamp without time zone	ends_at timestamp without time zone	
1	6	6	40000.000000000000	2023-10-04 20:02:43	2024-02-05 00:48:35	
2	9	9	42857.142857142857	2023-10-21 03:57:43	2023-12-28 13:24:27	
3	12	12	34285.714285714286	2023-10-12 14:38:33	2023-10-22 16:31:25	
4	14	14	35238.095238095238	2023-10-19 06:41:48	2024-05-16 02:14:46	

- Функция для получения информации о свободных одноместных номерах отеля на завтрашний день

CREATE OR REPLACE FUNCTION get_free_room_for_tomorrow(hotel_code_param
INTEGER)

RETURNS TABLE (

```

        code BIGINT,

        room_type_code BIGINT,

        number INTEGER,

        occupation_status VARCHAR(15),

        cleaning_status VARCHAR(15),

        hotel_code BIGINT

    )

LANGUAGE plpgsql

AS $$

BEGIN

    RETURN QUERY (

        SELECT *

        FROM room

        WHERE room.hotel_code = hotel_code_param

            AND room.occupation_status = 'free'

            AND room.code NOT IN (

                SELECT ac.room_code

                FROM accomodation_contract ac

                WHERE ac.booking_date <= CURRENT_DATE + 1

            )

    );

END;

$$

```

Query

Query History

1

SELECT *

2

FROM get_free_room_for_tomorrow(1);

Data Output

Messages

Notifications

≡

📄

▼

📋

▼

🗑

🗄

⬇

📈

	code bigint	room_type_code bigint	number integer	occupation_status character varying	cleaning_status character varying	hotel_code bigint
1	10	3	109	free	not cleaned	1
2	19	1	118	free	cleaned	1
3	23	9	202	free	not cleaned	1
4	44	7	303	free	cleaned	1
5	55	8	314	free	not cleaned	1
6	67	6	406	free	not cleaned	1
7	86	6	505	free	not cleaned	1

```
SELECT * FROM get_free_room_for_tomorrow(1);
```

- Процедура для бронирования двухместного номера в гостинице на заданную дату и количество дней проживания

```
DROP PROCEDURE IF EXISTS book_room_for_two_people;
```

```
CREATE PROCEDURE book_room_for_two_people(
```

```
    hotel_code_param INTEGER,
```

```
    guest_id_param INTEGER,
```

```
    arrival_date_param DATE,
```

```
    days_to_stay INTEGER
```

```
)
```

```
LANGUAGE plpgsql
```

```
AS $$
```

```
DECLARE
```

```
    target_room_code INTEGER;
```

```
    calculated_final_price NUMERIC;
```

```
BEGIN
```

```
    IF (
```

```
        SELECT COUNT(*) FROM room r
```

```

JOIN room_type rt

        ON r.room_type_code = rt.code

WHERE r.hotel_code = 1

        AND r.occupation_status = 'free'

        AND rt.guests_amount = 2

) > 0

THEN

SELECT r.code, p.base_price

INTO target_room_code, calculated_final_price

FROM room r

JOIN room_type rt

        ON r.room_type_code = rt.code

JOIN price p

        ON p.room_type_code = rt.code

WHERE r.hotel_code = 1

        AND r.occupation_status = 'free'

        AND rt.guests_amount = 2

ORDER BY p.base_price

LIMIT 1;

INSERT INTO accomodation_contract (

        guest_id,

        room_code,

        state,

        booking_date,

        arrival_date,

        departure_date,

```



```

        payment_state,

        employee_attendance_id,

        final_price
    )

VALUES (

        guest_id_param,

        target_room_code,

        'Активен',

        CURRENT_DATE,

        arrival_date_param,

        arrival_date_param + days_to_stay,

        'reserved',

        1,

        calculated_final_price * days_to_stay

    );

END IF;

END;

$$

CALL book_room_for_two_people(1, 1, '2023-12-05', 5);

```

Query

Query History

Scratch Pad

1SELECT * FROM public.accomodation_contract

2ORDER BY "number" ASC

Data Output

Messages

Notifications

number

[PK] bigint

guest_id

bigint

room_code

bigint

state

character varying (100)

booking_date

date

arrival_date

date

departure_date

date

payment_state

character varying (15)

details

character varying (1000)

83

83

32

58

Все хорошо

2023-08-22

2023-09-07

2023-10-24

not paid

Написать экзамен исполнять да выгна

84

84

24

95

Все хорошо

2023-01-17

2023-04-05

2023-05-14

declined

Аж неправда механический ягода возм

85

85

50

20

Все хорошо

2023-09-02

2023-09-23

2023-10-25

paid

Приходить роскошный выкинуть преле

86

86

52

8

Все хорошо

2023-05-07

2023-07-04

2023-09-29

declined

Издали спешить выдержать заплакать

87

87

51

37

Все хорошо

2023-07-17

2023-07-26

2023-10-27

not paid

Промолчать виднеться крутой устройс

88

88

27

87

Все хорошо

2023-03-19

2023-07-06

2023-08-06

not paid

Оборот казнь единый выраженный раз

89

89

72

62

Все хорошо

2023-09-09

2023-09-18

2023-10-03

not paid

Столетие присест заработать деловой

90

90

52

17

Все хорошо

2023-04-12

2023-08-02

2023-09-04

reserved

Вскинуть мальчишка растеряться мест

91

91

12

49

Все хорошо

2023-02-18

2023-05-03

2023-05-06

paid

Растеряться заложить покинуть запла

92

92

73

13

Все хорошо

2023-01-05

2023-01-08

2023-02-14

not paid

Художественный порог освобождение

93

93

70

9

Все хорошо

2023-06-15

2023-08-18

2023-09-14

declined

Угроза наслаждение покинуть оборот €

94

94

91

87

Все хорошо

2023-02-01

2023-06-02

2023-08-13

reserved

Отражение изредка миг через потрясти

95

95

66

100

Все хорошо

2023-02-10

2023-03-26

2023-05-18

reserved

Оборот ломать упорно бегать жидкий.

96

96

47

2

Все хорошо

2023-01-20

2023-09-01

2023-10-12

paid

Вряд рот покинуть зарплата вчера дых

97

97

95

27

Все хорошо

2023-03-06

2023-03-19

2023-10-08

declined

Запретить инструкция шлем багровый

98

98

61

62

Все хорошо

2023-05-21

2023-06-02

2023-06-28

declined

Сохранять ленинград коричневый. Лом

99

99

81

59

Все хорошо

2023-07-23

2023-09-29

2023-10-16

declined

Набор дыхание коллектив ведь. Миг эк

100

100

74

5

Все хорошо

2023-02-03

2023-11-04

2023-11-30

reserved

Миллиард запустить прежний налево.

Query

Query History

Scratch Pad

1

SELECT * FROM public.accomodation_contract

2

ORDER BY "number" ASC

Data Output

Messages

Notifications

	number [PK] bigint	guest_id bigint	room_code bigint	state character varying (100)	booking_date date	arrival_date date	departure_date date	payment_state character varying (15)	details character varying (1000)
84	84	24	95	Все хорошо	2023-01-17	2023-04-05	2023-05-14	declined	Аж неправда механический ягода возм
85	85	50	20	Все хорошо	2023-09-02	2023-09-23	2023-10-25	paid	Приходить роскошный выкинуть преле
86	86	52	8	Все хорошо	2023-05-07	2023-07-04	2023-09-29	declined	Издали спешить выдержать заплакать
87	87	51	37	Все хорошо	2023-07-17	2023-07-26	2023-10-27	not paid	Промолчать виднеться крутой устройс
88	88	27	87	Все хорошо	2023-03-19	2023-07-06	2023-08-06	not paid	Оборот казнь единый выраженный раз
89	89	72	62	Все хорошо	2023-09-09	2023-09-18	2023-10-03	not paid	Столетие присест заработать деловой
90	90	52	17	Все хорошо	2023-04-12	2023-08-02	2023-09-04	reserved	Вскинуть мальчишка растеряться мест
91	91	12	49	Все хорошо	2023-02-18	2023-05-03	2023-05-06	paid	Растеряться заложить покинуть заплак
92	92	73	13	Все хорошо	2023-01-05	2023-01-08	2023-02-14	not paid	Художественный порог освобождение :
93	93	70	9	Все хорошо	2023-06-15	2023-08-18	2023-09-14	declined	Угроза наслаждение покинуть оборот б
94	94	91	87	Все хорошо	2023-02-01	2023-06-02	2023-08-13	reserved	Отражение изредка миг через потрясти
95	95	66	100	Все хорошо	2023-02-10	2023-03-26	2023-05-18	reserved	Оборот ломать упорно бегать жидкий. f
96	96	47	2	Все хорошо	2023-01-20	2023-09-01	2023-10-12	paid	Вряд рот покинуть зарплата вчера дых
97	97	95	27	Все хорошо	2023-03-06	2023-03-19	2023-10-08	declined	Запретить инструкция шлем багровый i
98	98	61	62	Все хорошо	2023-05-21	2023-06-02	2023-06-28	declined	Сохранять ленинград коричневый. Лом
99	99	81	59	Все хорошо	2023-07-23	2023-09-29	2023-10-16	declined	Набор дыхание коллектив ведь. Миг эк
100	100	74	5	Все хорошо	2023-02-03	2023-11-04	2023-11-30	reserved	Миллиард запустить прежний налево. T
101	107	1	57	Активен	2023-12-04	2023-12-05	2023-12-10	reserved	[null]

4. Модифицировать триггер (триггерную функцию) на проверку корректности входа и выхода сотрудника

-- false - enter shift

-- true - exit shift

create or replace function fn_check_time_punch() returns trigger as \$psql\$

begin

```

if new.is_out_punch = (
    select tps.is_out_punch
    from time_punch tps
    where tps.employee_id = new.employee_id
    order by tps.id desc limit 1
) or new.employee_id != (
    select tps.employee_id
    from time_punch tps
    order by tps.id desc limit 1
) or new.punch_time <= (
    select tps.punch_time
    from time_punch tps
    where tps.employee_id = new.employee_id
    order by tps.id desc limit 1
)
then
    return null;
end if;

return new;

end;

$sql$ language plpgsql;

DROP TRIGGER IF EXISTS check_time_punch ON time_punch;

CREATE TRIGGER check_time_punch
BEFORE INSERT ON time_punch
FOR EACH ROW

```

```
EXECUTE PROCEDURE fn_check_time_punch();
```

5. Авторский триггер для обновления статуса занятости номера в отеле при создании нового договора о проживании

```
CREATE OR REPLACE FUNCTION change_room_occupation_status()
```

```
RETURNS TRIGGER AS $$
```

```
BEGIN
```

```
    UPDATE room
```

```
    SET occupation_status = 'booked'
```

```
    WHERE code = NEW.room_code;
```

```
    RETURN NEW;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER change_room_occupation_status
```

```
AFTER INSERT
```

```
ON accomodation_contract
```

```
FOR EACH ROW
```

```
EXECUTE FUNCTION change_room_occupation_status();
```

Вывод

В ходе выполнения данной лабораторной работы я научился создавать и работать с процедурами, функциями и триггерами в PostgreSQL.