Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ»

Факультет среднего профессионального образования

**ОТЧЕТ**

**о Лабораторной работе № 4**

по теме: ПРЕДСТАВЛЕНИЯ**.** ХРАНИМЫЕ ПРОЦЕДУРЫ И ФУНКЦИИ. ТРИГГЕРЫ

по дисциплине: Разработка, администрирование и защита баз данных

Специальность:

09.02.07 Информационные системы и программирование

УП.11 Разработка, администрирование и защита баз данных

|  |  |
| --- | --- |
| Проверил:  Ефимова Т.Н. \_\_\_\_\_\_\_\_\_\_  Дата: «\_\_» \_\_\_\_\_\_\_ 2019г.  Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Выполнил:  студент группы Y2336  \_\_\_\_\_\_ Заерко М.В. |

Санкт-Петербург 2019

1. Цель работы

Формирование умений и навыков для работы в среде СУБД MySQL. Получить практические навыки по программированию в СУБД MySQL.

1. Практическое задание

Практическое задание состоит из следующих пунктов:

1. Создать представления для выполнения запросов.
2. Создать процедуры и функции для ограничения целостности и выполнения запросов с параметрами (для ограничения целостности и выполнения запросов.
3. Создать триггеры на все события, триггеры должны быть на вставку, удаление, редактирование не менее 3-х.
4. Выполнение лабораторной работы
5. Скрипты создания представлений, процедур, функций, триггеров

На листингах 1- представлены скрипты для создания процедур, триггеров и функций к базе данных.

Листинг 1 – Определяет, когда самолету необходим ремонт

DELIMITER //

CREATE TRIGGER need\_repair

BEFORE INSERT

ON itog.plane

FOR EACH ROW

BEGIN

update itog.plane set rapared = 'need'

where (select timestampdiff(year, date\_produce ,now()))>=10

AND (select timestampdiff(year, date\_last\_repair,now()))>=1;

END

//

DELIMITER ;

Листинг 2 – Определяет, можно ли купить билет

DELIMITER //

CREATE TRIGGER can\_buy

BEFORE INSERT

ON itog.flight

FOR EACH ROW

BEGIN

update itog.flight set can\_buy = 'no'

where (select timestampdiff(day, now(), date\_flight))>=30

AND (select timestampdiff(day, now(), date\_flight))<=1;

END

//

DELIMITER ;

Листинг 3 – Определяет, сколько будет длиться полет

DELIMITER //

CREATE FUNCTION time\_of\_flight ( flightt INT )

RETURNS FLOAT

BEGIN

DECLARE t FLOAT;

DECLARE s FLOAT;

DECLARE v INT;

SET s = (SELECT distanse FROM itog.flight WHERE id\_flight = flightt);

SET v = (SELECT speed FROM itog.plane WHERE bort\_number = (SELECT plane\_bort\_number FROM itog.flight WHERE id\_flight = flightt));

SET t = s/v;

RETURN t;

END; //

DELIMITER ;

;

Листинг 5 – Функция, считает расход топлива для полета

DELIMITER //

CREATE FUNCTION wasted\_patrol ( flightt INT )

RETURNS FLOAT

BEGIN

DECLARE t FLOAT;

DECLARE s INT;

DECLARE v FLOAT;

SET t = time\_of\_flight(flightt);

SET s = (SELECT consumption FROM itog.plane WHERE bort\_number = (SELECT plane\_bort\_number FROM itog.flight WHERE id\_flight = flightt));

SET v = t\*s;

RETURN v;

END; //

DELIMITER ;

Листинг 8 – Функция, считает количество свободных мест на рейсе.

DELIMITER //

CREATE FUNCTION free\_places ( flightt INT )

RETURNS FLOAT

BEGIN

DECLARE num\_places INT;

DECLARE pass INT;

DECLARE free INT;

SET pass = (SELECT COUNT(\*) FROM itog.passanger WHERE flight\_id\_flight = flightt);

SET num\_places = (SELECT places\_num FROM itog.plane WHERE bort\_number = (SELECT plane\_bort\_number FROM itog.flight WHERE id\_flight = flightt));

SET free = num\_places - pass;

RETURN free;

END; //

DELIMITER ;

Листинг 10 – Функция, считает свободные места бизнес класса по каждому рейсу

DELIMITER //

CREATE FUNCTION free\_business\_places ( flightt INT )

RETURNS INT

BEGIN

DECLARE num\_places\_business INT;

DECLARE pass\_business INT;

DECLARE free\_business INT;

SET pass\_business = (SELECT COUNT(\*) FROM itog.passanger WHERE flight\_id\_flight = flightt AND type\_place = 'Business');

SET num\_places\_business = (SELECT business FROM itog.plane WHERE bort\_number = (SELECT plane\_bort\_number FROM itog.flight WHERE id\_flight = flightt));

SET free\_business = num\_places\_business - pass\_business;

RETURN free\_business;

END; //

DELIMITER ;

Листинг 11 – Функция, считает свободные места эконом класса по каждому рейсу

DELIMITER //

CREATE FUNCTION free\_economy\_places ( flightt INT )

RETURNS INT

BEGIN

DECLARE num\_places\_economy INT;

DECLARE pass\_economy INT;

DECLARE free\_economy INT;

SET pass\_economy = (SELECT COUNT(\*) FROM itog.passanger WHERE flight\_id\_flight = flightt AND type\_place = 'Econom');

SET num\_places\_economy = (SELECT econom FROM itog.plane WHERE bort\_number = (SELECT plane\_bort\_number FROM itog.flight WHERE id\_flight = flightt));

SET free\_economy = num\_places\_economy - pass\_economy;

RETURN free\_economy;

END; //

DELIMITER ;

Листинг 12 – Функция, считает убытки по рейсу

DELIMITER //

CREATE FUNCTION wasted\_money ( flightt INT )

RETURNS INT

BEGIN

DECLARE price\_economy INT;

DECLARE price\_business INT;

DECLARE free\_economy INT;

DECLARE free\_business INT;

DECLARE wast INT;

SET price\_economy = (SELECT price\_econom FROM itog.flight WHERE id\_flight = flightt);

SET price\_business = (SELECT price\_business FROM itog.flight WHERE id\_flight = flightt);

SET free\_economy = (SELECT free\_economy\_places(flightt));

SET free\_business = (SELECT free\_business\_places(flightt));

SET wast = free\_business\*price\_business + free\_economy\*price\_economy;

RETURN wast;

END; //

DELIMITER ;

Листинг 18 – Показ доступных билетов

DELIMITER //

CREATE PROCEDURE show\_available\_tickets()

BEGIN

SELECT id\_flight, date\_flight,departure\_time, time\_of\_flight(id\_flight),airport\_departure, airport\_arrive, price\_business, price\_econom

FROM itog.flight WHERE can\_buy='yes';

END

//

DELIMITER ;

Листинг 19 – Бронирование билета

DELIMITER //

CREATE PROCEDURE buy\_ticket(id INT(11),type\_place varchar(45), passport INT(11), name VARCHAR(45))

BEGIN

DECLARE i INT;

DECLARE j INT;

DECLARE l INT;

SET i = (SELECT place\_num FROM itog.passanger WHERE flight\_id\_flight = id);

IF (i=null) THEN

SET j =(SELECT (id\_passanger+1) FROM itog.passanger ORDER BY id\_passanger DESC LIMIT 1);

INSERT INTO passanger VALUES (passport,name, 1, type\_place, id,j);

ELSE

SET j =(SELECT (id\_passanger+1) FROM itog.passanger ORDER BY id\_passanger DESC LIMIT 1);

SET l= (SELECT (place\_num+1) FROM itog.passanger WHERE flight\_id\_flight = id

ORDER BY place\_num DESC LIMIT 1);

INSERT INTO passanger VALUES (passport,name,l,type\_place, id, j);

END IF;

END

//

DELIMITER ;

Листинг 20 – Просмотр купленного билета

DELIMITER //

CREATE PROCEDURE show\_ticket(id INT(11))

BEGIN

SELECT A.fio\_pass, A.place\_num, A.type\_place, B.date\_flight, B.departure\_time, B.airport\_departure, B.airport\_arrive FROM itog.passanger A, itog.flight B

WHERE A.flight\_id\_flight = id AND B.id\_flight = id;

END

//

DELIMITER ;

1. Скриншоты работы представлений, процедур, функций, триггеров

Скриншоты работы процедур, функций и триггеров представлены на рисунках 1- 12.

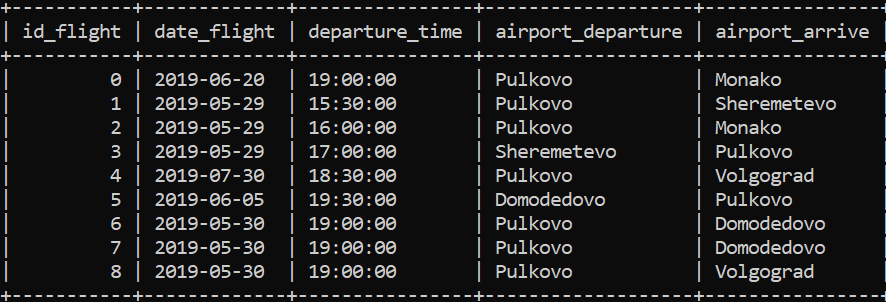


Рисунок 1 – Результат работы первого триггера

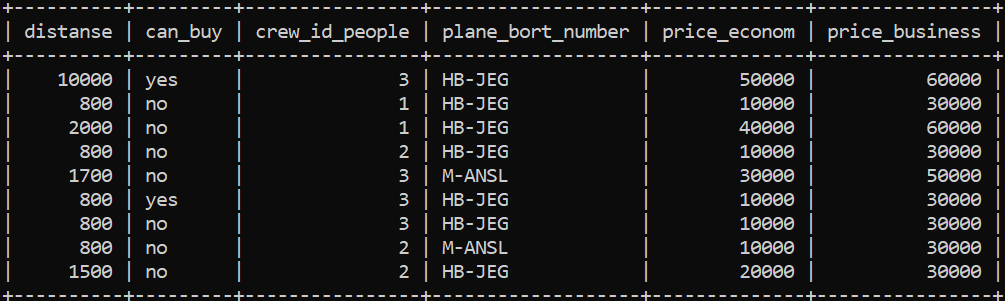


Рисунок 2 – Результат работы первого триггера

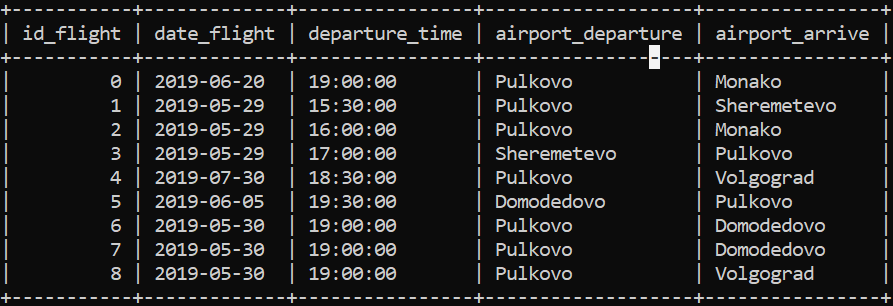


Рисунок 3 – Результат работы первого триггера

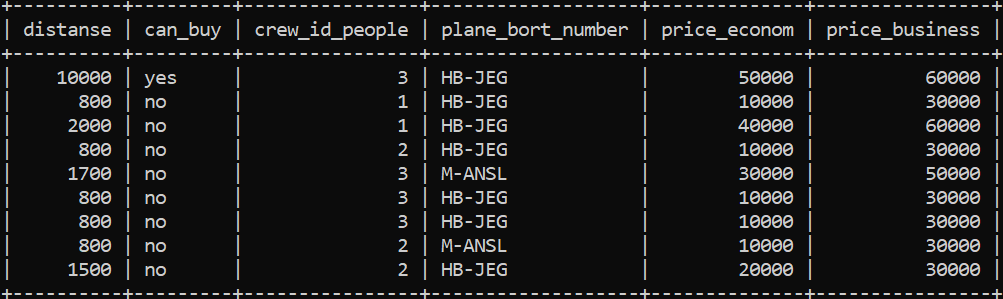


Рисунок 4 – Результат работы первого триггера

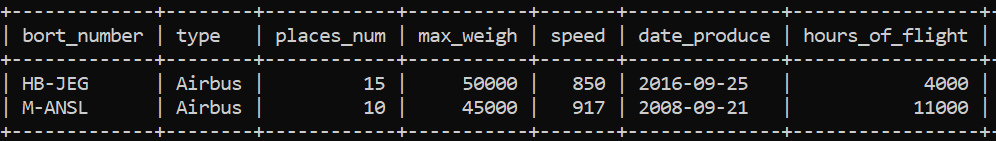


Рисунок 5 – Результат работы второго триггера

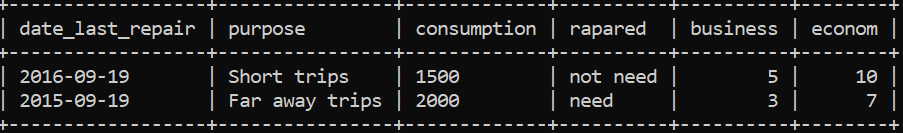


Рисунок 6 – Результат работы второго триггера

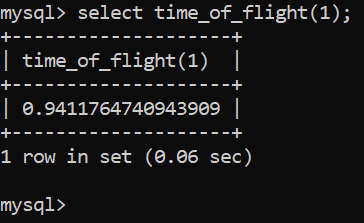


Рисунок 7 – Результат работы первой функции

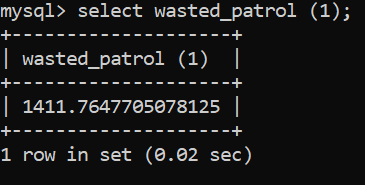


Рисунок 8 – Результат работы второй функции

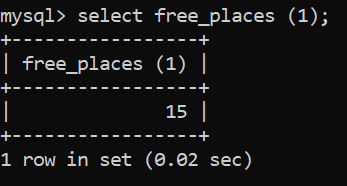


Рисунок 9 – Результат работы третьей функции

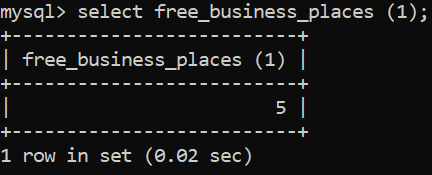


Рисунок 8 – Результат работы четвертой функции

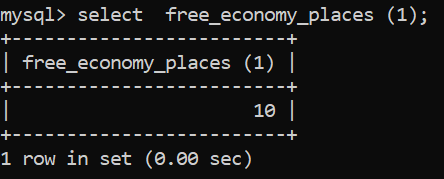


Рисунок 9 – Результат работы пятой функции

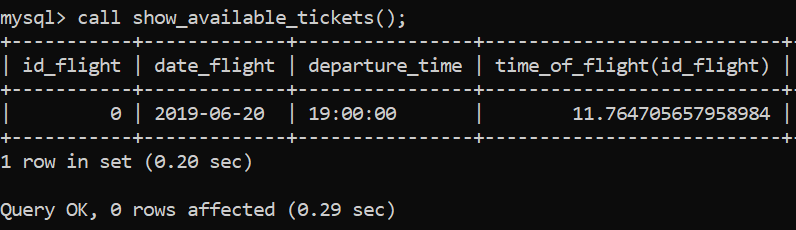


Рисунок 10 – Результат работы представления для пользователя

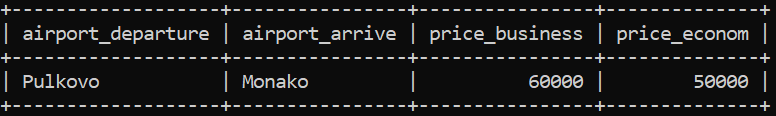


Рисунок 11 – Результат работы представления для пользователя

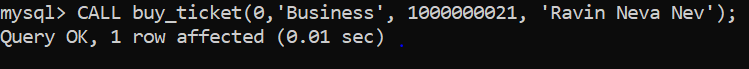


Рисунок 12 – Результат работы представления для пользователя

1. Вывод

В процессе выполнения лабораторной работы были приобретены практические навыки создания функций, процедур, представлений и триггеров для базы данных.