МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМЕНИ И.С. ТУРГЕНЕВА»

Кафедра информационных систем

ОТЧЕТ

По лабораторной работе №6

на тему: «Разработка хранимых процедур и триггеров»

по дисциплине: «Базы данных»

Выполнил: Бородин А. А.

Институт приборостроения, автоматизации и информационных технологий

Направление: 09.03.04 «Программная инженерия»

Группа: 91ПГ

Проверил: Рыженков Д. В.

Отметка о зачете:

Дата: «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2021 г.

Орел, 2021

1. Студент может быть старостой только в своей группе.

DELIMITER //

CREATE TRIGGER headmanCheckTrigger

BEFORE UPDATE

ON Группа

FOR EACH ROW

IF NEW.Староста\_зачетка NOT IN (SELECT Зачетка FROM Студент WHERE Группа\_id = NEW.id) THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Нельзя назначить старостой студента, не принадлежащего группе.';

END IF//

DELIMITER ;

1. Год рождения должен быть меньше года поступления.

DELIMITER //

CREATE TRIGGER BirthDateCheckOnUpdateTrigger

BEFORE UPDATE

ON Студент

FOR EACH ROW

IF TIMESTAMPDIFF(YEAR, NEW.Дата\_рождения, NEW.Год\_поступления) < 0 THEN

SIGNAL SQLSTATE '45001'

SET MESSAGE\_TEXT = 'Ошибка: год рождения больше года поступления';

END IF//

DELIMITER ;

DELIMITER //

CREATE TRIGGER BirthDateCheckOnInsertTrigger

BEFORE INSERT

ON Студент

FOR EACH ROW

IF TIMESTAMPDIFF(YEAR, NEW.Дата\_рождения, NEW.Год\_поступления) < 0 THEN

SIGNAL SQLSTATE '45001'

SET MESSAGE\_TEXT = 'Ошибка: год рождения больше года поступления';

END IF//

DELIMITER ;

1. Номер телефона может начинаться только с 8.

DELIMITER //

CREATE TRIGGER phoneNumberCheckOnUpdateTrigger

BEFORE UPDATE

ON Студент

FOR EACH ROW

IF LEFT(New.Номер\_телефона, 1) != '8' THEN

SIGNAL SQLSTATE '45002'

SET MESSAGE\_TEXT = 'Ошибка: номер телефона начинается не с 8';

END IF//

DELIMITER ;

DELIMITER //

CREATE TRIGGER phoneNumberCheckOnInsertTrigger

BEFORE INSERT

ON Студент

FOR EACH ROW

IF LEFT(New.Номер\_телефона, 1) != '8' THEN

SIGNAL SQLSTATE '45002'

SET MESSAGE\_TEXT = 'Ошибка: номер телефона начинается не с 8';

END IF//

DELIMITER ;

1. Номер курса должен находиться в промежутке с 1 до 5.

DELIMITER //

CREATE TRIGGER courseNumberCheckOnUpdateTrigger

BEFORE UPDATE

ON План\_дисциплин

FOR EACH ROW

IF NEW.Курс NOT BETWEEN 1 AND 5 THEN

SIGNAL SQLSTATE '45010'

SET MESSAGE\_TEXT = 'Ошибка: номер курса не принадлежит домену [1; 5]';

END IF//

DELIMITER ;

DELIMITER //

CREATE TRIGGER courseNumberCheckOnInsertTrigger

BEFORE INSERT

ON План\_дисциплин

FOR EACH ROW

IF NEW.Курс NOT BETWEEN 1 AND 5 THEN

SIGNAL SQLSTATE '45010'

SET MESSAGE\_TEXT = 'Ошибка: номер курса не принадлежит домену [1; 5]';

END IF//

DELIMITER ;

1. Номер семестра должен находится в промежутке с 1 до 10.

DELIMITER //

CREATE TRIGGER semesterNumberCheckOnUpdateTrigger

BEFORE UPDATE

ON План\_дисциплин

FOR EACH ROW

IF NEW.Номер\_семестра NOT BETWEEN 1 AND 10 THEN

SIGNAL SQLSTATE '45003'

SET MESSAGE\_TEXT = 'Ошибка: номер семестра не принадлежит домену [1; 10]';

END IF//

DELIMITER ;

DELIMITER //

CREATE TRIGGER semesterNumberCheckOnInsertTrigger

BEFORE INSERT

ON План\_дисциплин

FOR EACH ROW

IF NEW.Номер\_семестра NOT BETWEEN 1 AND 10 THEN

SIGNAL SQLSTATE '45004'

SET MESSAGE\_TEXT = 'Ошибка: номер семестра не принадлежит домену [1; 10] ';

END IF//

DELIMITER ;

1. Название типа занятие может быть только лекция, практика или лабораторная работа.

DELIMITER //

CREATE TRIGGER lessonTypeRestrictionOnInsertTrigger

BEFORE INSERT

ON Тип\_занятия

FOR EACH ROW

IF NEW.Название NOT IN ('Лекция', 'Лабораторная работа', 'Практика') THEN

SIGNAL SQLSTATE '45005'

SET MESSAGE\_TEXT = 'Ошибка: неверное название типа занятия';

END IF//

DELIMITER ;

DELIMITER //

CREATE TRIGGER lessonTypeRestrictionOnUpdateTrigger

BEFORE UPDATE

ON Тип\_занятия

FOR EACH ROW

IF NEW.Название NOT IN ('Лекция', 'Лабораторная работа', 'Практика') THEN

SIGNAL SQLSTATE '45005'

SET MESSAGE\_TEXT = 'Ошибка: неверное название типа занятия';

END IF//

DELIMITER ;

1. Количество часов на тип занятия может быть 2 или 4 часа.

DELIMITER //  
 CREATE TRIGGER CountInLesson  
 BEFORE INSERT  
 ON тип\_занятия  
 FOR EACH ROW  
 IF NEW.Длительность != 2 or NEW.Длительность != 4 THEN  
 SIGNAL SQLSTATE '45000'  
 SET MESSAGE\_TEXT = 'Введено некорректное число длительность занятия';  
 END IF//  
 DELIMITER ;  
   
 DELIMITER //  
 CREATE TRIGGER CountInLesson  
 BEFORE UPDATE  
 ON тип\_занятия  
 FOR EACH ROW  
 IF NEW.Длительность != 2 or NEW.Длительность != 4 THEN  
 SIGNAL SQLSTATE '45000'  
 SET MESSAGE\_TEXT = 'Введено некорректное число длительность занятия';  
 END IF//  
 DELIMITER ;

1. Количество часов на план дисциплин должно быть больше, чем количество часов на тип занятия. И общее количество часов на дисциплину должно быть кратно 2.

DELIMITER //  
 CREATE TRIGGER CountOfAllHrsMoreThenCurrentTypeOfLesson  
 BEFORE INSERT  
 ON план\_дисциплин  
 FOR EACH ROW  
 IF (NEW.Запланированно\_часов < (SELECT Длительность FROM Тип\_занятия WHERE Тип\_занятия.id = NEW.id) or NEW.Запланированно\_часов % 2 != 0) THEN  
 SIGNAL SQLSTATE '45000'  
 SET MESSAGE\_TEXT = 'Введено некорректно число на план дисциплины';  
 END IF//  
 DELIMITER ;  
   
 DELIMITER //  
 CREATE TRIGGER CountOfAllHrsMoreThenCurrentTypeOfLesson  
 BEFORE UPDATE  
 ON план\_дисциплин  
 FOR EACH ROW  
 IF (NEW.Запланированно\_часов < (SELECT Длительность FROM Тип\_занятия WHERE Тип\_занятия.id = NEW.id) or NEW.Запланированно\_часов % 2 != 0) THEN  
 SIGNAL SQLSTATE '45000'  
 SET MESSAGE\_TEXT = 'Введено некорректно число на план дисциплины';  
 END IF//  
 DELIMITER ;

1. Посещение студентов отмечается только на занятиях их группы.

DELIMITER //  
 CREATE TRIGGER StudientOnHisLesson  
 BEFORE INSERT  
 ON посещаемость  
 FOR EACH ROW  
 IF (NEW.Занятие\_Группа\_id != (SELECT Группа\_id FROM Студент WHERE Группа.id = NEW.id)) THEN  
 SIGNAL SQLSTATE '45000'  
 SET MESSAGE\_TEXT = 'Студент не находится в этой группе';  
 END IF//  
 DELIMITER ;  
   
 DELIMITER //  
 CREATE TRIGGER StudientOnHisLesson  
 BEFORE UPDATE  
 ON посещаемость  
 FOR EACH ROW  
 IF (NEW.Занятие\_Группа\_id != (SELECT Группа\_id FROM Студент WHERE Группа.id = NEW.id)) THEN  
 SIGNAL SQLSTATE '45000'  
 SET MESSAGE\_TEXT = 'Студент не находится в этой группе';  
 END IF//  
 DELIMITER ;

Процедуры

1. Вывести количество студентов в группе по шифру

DELIMITER //

CREATE PROCEDURE count\_students\_in\_group(code VARCHAR(30))

BEGIN

SELECT COUNT(Студент.Зачетка) FROM Студент

INNER JOIN Группа ON Студент.Группа\_id = Группа.id;

END//

1. Вывести имена всех старост

DELIMITER //

CREATE PROCEDURE print\_headMen()

BEGIN

SELECT Студент.Зачетка, Студент.Фамилия, Студент.Имя, Группа.Зачетка FROM Студент

RIGHT JOIN Группа ON Студент.Группа\_id = Группа.id

WHERE Студент.Зачетка = Группа.Староста\_зачетка;

END//

1. Вывести количество пар по предмету

DELIMITER //

CREATE PROCEDURE count\_lessons(lesson\_name VARCHAR(100))

BEGIN

SELECT Дисциплина.Название, Тип\_занятия.Название, COUNT(1) FROM Занятие

JOIN План\_дисциплин ON План\_дисциплин.id = Занятие.План\_дисциплин\_id

JOIN Дисциплина ON Дисциплина.id = План\_дисциплин.Дисциплина\_id

JOIN Тип\_занятия ON Тип\_занятия.id = План\_дисциплин.Тип\_занятия\_id

GROUP BY Дисциплина.Название, Тип\_занятия.Название HAVING Дисциплина.Название = lesson\_name;

END//