/\* Задание 1. Создайте хранимую процедуру, которая создает временную таблицу для подсчета из каких городов пользователи в бд. \*/

DROP PROCEDURE IF EXISTS user\_city;

DELIMITER //

CREATE PROCEDURE user\_city()

BEGIN

SELECT p.hometown, count(p.user\_id) AS user

FROM users u

JOIN profiles p ON p.user\_id = u.id

GROUP BY p.hometown ;

END //

DELIMITER ;

CALL user\_city();

/\* Задание 2. Создайте триггер, который при удалении пользователя удаляет также его файлы \*/

DROP TRIGGER IF EXISTS media\_deletion;

DELIMITER //

CREATE TRIGGER media\_deletion AFTER DELETE ON users

FOR EACH ROW

BEGIN

DELETE FROM media m WHERE m.user\_id = OLD.id;

END //

DELIMITER ;

/\* Задание 3. Создайте триггер, который при добавлении пользователя проверяет его дату рождения и устанавливает текущую дату, если дата из "будущего" \*/

DROP TRIGGER IF EXISTS date\_insert;

DELIMITER //

CREATE TRIGGER date\_insert BEFORE INSERT ON profiles

FOR EACH ROW

BEGIN

IF NEW.birthday > curdate() THEN

SET NEW.birthday = curdate();

END IF;

END //

DELIMITER ;

/\* Задание 4. Создайте триггер, который при обновлении данных в таблице media будет ставить текущую дату и время в колонку updated\_at \*/

DROP TRIGGER IF EXISTS media\_update;

DELIMITER //

CREATE TRIGGER media\_update BEFORE UPDATE ON media

FOR EACH ROW

BEGIN

SET NEW.updated\_at = curdate();

END //

DELIMITER ;

/\* Задание 5. Создайте триггер, который проверяет на правильность ввод данных о пользователе

при вставке нового пользователя ( fristname и lastname, email не должны быть пустыми, phone начинается с 7), и выводит на экран ошибку "Invalid user data" \*/

DROP TRIGGER IF EXISTS correct\_date;

DELIMITER //

CREATE TRIGGER correct\_date BEFORE UPDATE ON users

FOR EACH ROW

IF (NEW.firstname AND NEW.lastname AND NEW.email) IS NOT NULL AND (NEW.phone < 70000000000) THEN

SIGNAL SQLSTATE '45000' -- Ошибка с кодом 45000

SET MESSAGE\_TEXT = 'Invalid user date'; -- Сообщение об ошибке

END IF //

DELIMITER ;

/\* Задание 6. Создайте функцию, которая удаляет пользователя по id вместе с его профилем \*/

DROP FUNCTION IF EXISTS user\_del\_byid;

DELIMITER //

CREATE FUNCTION user\_del\_byid(del\_user\_id int)

RETURNS int READS SQL DATA NOT DETERMINISTIC

BEGIN

DELETE FROM profiles WHERE user\_id = del\_user\_id;

DELETE FROM users WHERE id = del\_user\_id;

RETURN del\_user\_id;

END //

DELIMITER ;

/\* Задание 7. Создайте функцию hello(), которая будет возвращать приветствие, в зависимости от текущего времени суток.

С 6:00 до 12:00 функция должна возвращать фразу "Доброе утро",

с 12:00 до 18:00 функция должна возвращать фразу "Добрый день",

с 18:00 до 00:00 — "Добрый вечер",

с 00:00 до 6:00 — "Доброй ночи" \*/

DROP FUNCTION IF EXISTS hello;

DELIMITER //

CREATE FUNCTION hello()

RETURNS VARCHAR(50) DETERMINISTIC

BEGIN

DECLARE result\_text VARCHAR(25);

SELECT CASE

WHEN CURRENT\_TIME >= '06:00:00' AND CURRENT\_TIME < '12:00:00' THEN 'Доброе утро'

WHEN CURRENT\_TIME >= '12:00:00' AND CURRENT\_TIME < '18:00:00' THEN 'Добрый день'

WHEN CURRENT\_TIME >= '18:00:00' AND CURRENT\_TIME < '00:00:00' THEN 'Добрый вечер'

ELSE 'Доброй ночи'

END INTO result\_text;

RETURN result\_text;

END //

DELIMITER ;