1. В базе данных shop и sample присутствуют одни и те же таблицы, учебной базы данных. Переместите запись id = 1 из таблицы shop.users в таблицу sample.users. Используйте транзакции.

START TRANSACTION;

SELECT \* FROM shop.users WHERE id = 1;  
INSERT INTO sample.users SELECT \* FROM shop.users WHERE id = 1;  
DELETE FROM shop.users WHERE id = 1 LIMIT 1;

COMMIT;

1. Создайте представление, которое выводит название name товарной позиции из таблицы products и соответствующее название каталога name из таблицы catalogs.

use shop;

CREATE VIEW prod\_cat (product\_name, catalog\_id)  
AS SELECT products.name, catalogs.name FROM products JOIN catalogs

ON products.catalog\_id = catalogs.id;  
SELECT \* FROM prod\_cat;

1. по желанию) Пусть имеется таблица с календарным полем created\_at. В ней размещены разряженые календарные записи за август 2018 года '2018-08-01', '2016-08-04', '2018-08-16' и 2018-08-17. Составьте запрос, который выводит полный список дат за август, выставляя в соседнем поле значение 1, если дата присутствует в исходном таблице и 0, если она отсутствует.

Не уверена, что такой вариант работает.

CREATE TABLE table\_2 (

day DATE,

day\_coincidence BOOLEAN);

DELIMITER //

CREATE PROCEDURE fill\_in\_dates ()

BEGIN

DECLARE i INT DEFAULT 31;

DECLARE tag DATE DEFAULT '2018-08-01';

WHILE i > 0 DO

INSERT INTO table\_2 (day, day\_coincidence) VALUES

(tag, IF (tag IN table\_1 JOIN table\_2, 1, 0))

SET tag = DATE\_ADD (tag, INTERVAL 1 DAY)

SET i = i - 1;

END WHILE;

END//

CALL fill\_in\_dates () //

1. (по желанию) Пусть имеется любая таблица с календарным полем created\_at. Создайте запрос, который удаляет устаревшие записи из таблицы, оставляя только 5 самых свежих записей.

Не уверена, что такой вариант работает:

PREPARE upd FROM ' DELETE \* FROM table\_1 WHERE created\_at NOT IN (SELECT \* FROM table\_1 ORDER BY created\_at DESC LIMIT 5) ';

EXECUTE upd;

1. Создайте хранимую 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".

DELIMITER //

DROP FUNCTION IF EXISTS hello() //

CREATE FUNCTION hello()  
RETURNS tinytext NO SQL  
BEGIN  
 DECLARE time TIME;  
 DECLARE greeting VARCHAR(50);  
 SET time = TIME(NOW());  
 SET greeting = ' ';  
 case  
 when time BETWEEN '6:00:00' AND '12:00:00' THEN  
 SET greeting = 'Доброе утро';  
 when time BETWEEN '12:00:00' AND '18:00:00' THEN  
 SET greeting = 'Добрый день';  
 when time BETWEEN '18:00:00' AND '00:00:00' THEN  
 SET greeting = 'Добрый вечер';  
 when time BETWEEN '00:00:00' AND '6:00:00' THEN  
 SET greeting = 'Доброй ночи';  
 END case;  
 RETURN greeting;  
END //

1. В таблице products есть два текстовых поля: name с названием товара и description с его описанием. Допустимо присутствие обоих полей или одно из них. Ситуация, когда оба поля принимают неопределенное значение NULL неприемлема. Используя триггеры, добейтесь того, чтобы одно из этих полей или оба поля были заполнены. При попытке присвоить полям NULL-значение необходимо отменить операцию.

Решить не получилось

DROP TRIGGER IF EXISTS check\_products\_update //

CREATE TRIGGER check\_products\_update BEFORE **UPDATE** ON products

FOR EACH ROW

BEGIN

IF NEW.name = NULL AND OLD.description = NULL THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Enter at least one name or description';

ELSEIF NEW.description = NULL AND OLD.name = NULL THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Enter at least one name or description';

ELSEIF NEW.name = NULL AND NEW.description = NULL THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Enter at least one name or description';

END IF;

END //

DROP TRIGGER IF EXISTS check\_products\_update //

CREATE TRIGGER check\_products\_update BEFORE UPDATE ON products

FOR EACH ROW

BEGIN

DECLARE line\_id INT;

IF name = NULL

SELECT id INTO line\_id FROM products WHERE name = NULL;

CASE WHEN description = NULL WHERE

AND description = NULL THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Enter at least one name or description';

END IF;

END //

CREATE TRIGGER check\_products\_update BEFORE **UPDATE** ON products

FOR EACH ROW

BEGIN

IF name = NULL AND OLD.description = NULL THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Enter at least one name or description';

ELSEIF description = NULL AND OLD.name = NULL THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Enter at least one name or description';

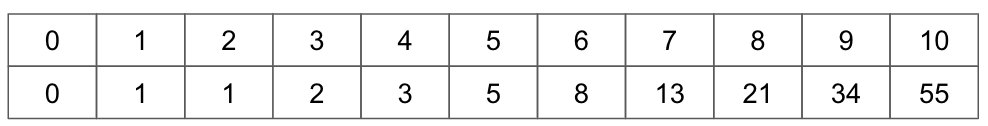
ELSEIF name = NULL AND description = NULL THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Enter at least one name or description';

END IF;

END //

1. (по желанию) Напишите хранимую функцию для вычисления произвольного числа Фибоначчи. Числами Фибоначчи называется последовательность в которой число равно сумме двух предыдущих чисел. Вызов функции FIBONACCI(10) должен возвращать число 55.



DROP PROCEDURE IF EXISTS FIBONACCI //

CREATE procedure FIBONACCI(number INT)

BEGIN

DECLARE i INT DEFAULT 3;

DECLARE num\_1 INT DEFAULT 1;

DECLARE num\_2 INT DEFAULT 1;

DECLARE result INT DEFAULT 0;

IF (number > 2) THEN

cycle : WHILE i <= number DO

SET result = num\_1 + num\_2;

SET num\_1 = num\_2;

SET num\_2 = result;

SET i = i + 1;

END WHILE cycle;

SELECT result;

ELSEIF (number > 0) THEN

SELECT '1';

ELSEIF (number = 0) THEN

SELECT '0';

ELSE

SELECT 'Ошибочное значение параметра';

END IF;

END//