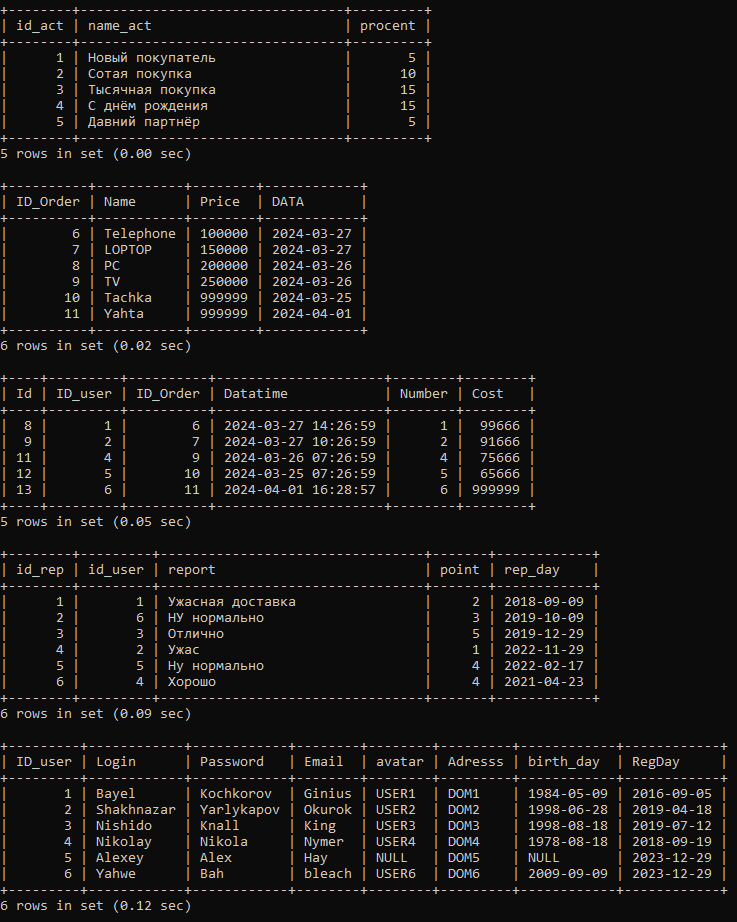
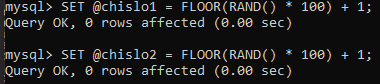
# Хранимый код



**Функции**

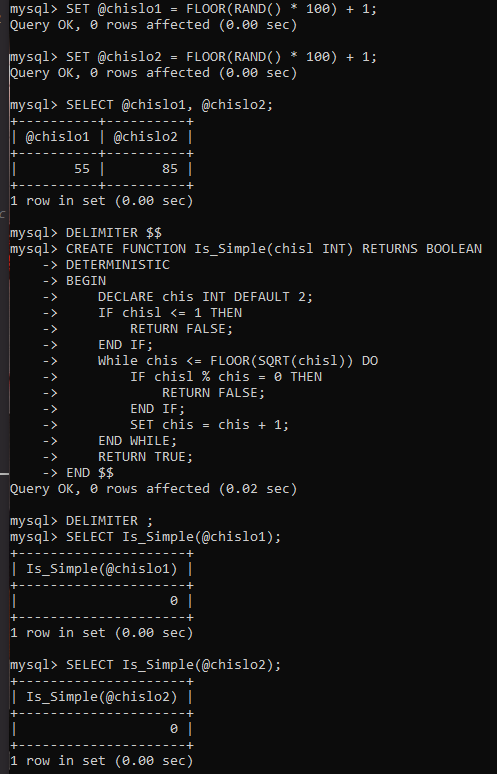
**Создать сеансовые переменные для двух случайных целых чисел от 1 до 100 (включительно).**

****

Оператор SET @chislo1/2 = FLOOR(RAND() \* 100) + 1; присваивает переменной @chislo1/2 случайное целое число в диапазоне от 1 до 100 (включая 1 и 100), генерируя случайное число от 0 до 1, умножая его на 100, округляя до ближайшего целого числа в меньшую сторону и добавляя 1.

**Создать функции:**

1. **Функцию Is\_Simple(), которая проверяет, является ли число простым (в ответе TRUE/FALSE).**

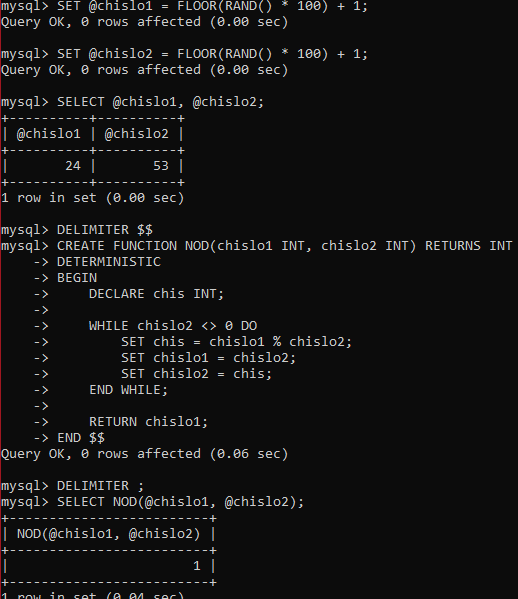
****

Функция Is\_Simple проверяет, является ли данное число chisl простым.

Она делает это путем проверки, делится ли chisl на какое-либо число от 2 до квадратного корня из chisl. Если chisl делится на какое-либо число в этом диапазоне, функция возвращает FALSE. В противном случае функция возвращает TRUE, что означает, что chisl является простым числом.

Функция является детерминированной, что означает, что она всегда будет возвращать один и тот же результат для данного входного значения chisl.

1. **Функцию NOD(), которая ищет наибольший общий делитель этих чисел (в ответе нужное целое число). Для вычисления используйте алгоритм Евклида.**

****

Этот код создает функцию с именем NOD, которая принимает два целых числа (chislo1 и chislo2) и возвращает их наибольший общий делитель (НОД).

Внутри функции используется цикл WHILE для вычисления НОД с помощью алгоритма Евклида. На каждой итерации цикла вычисляется остаток от деления chislo1 на chislo2, который затем присваивается переменной chis. Затем значения chislo1 и chislo2 обновляются, чтобы продолжить цикл до тех пор, пока chislo2 не станет равным 0.

После завершения цикла функция возвращает значение chislo1, которое и является НОД для входных чисел chislo1 и chislo2.

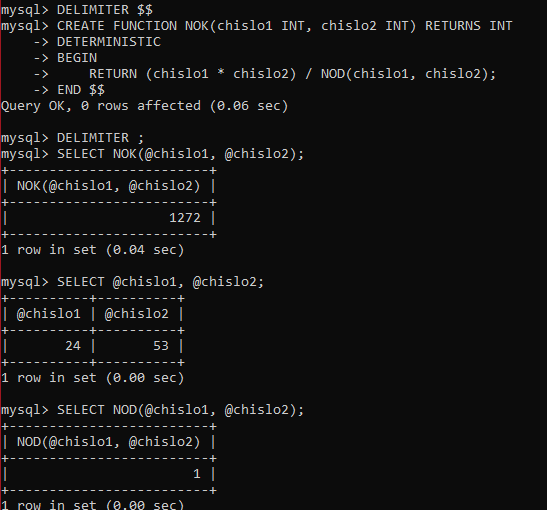
1. **Функцию NOK(), которая ищет наименьшее общее кратное этих чисел (в ответе нужное число). Для вычисления использовать функцию из пункта 2.**

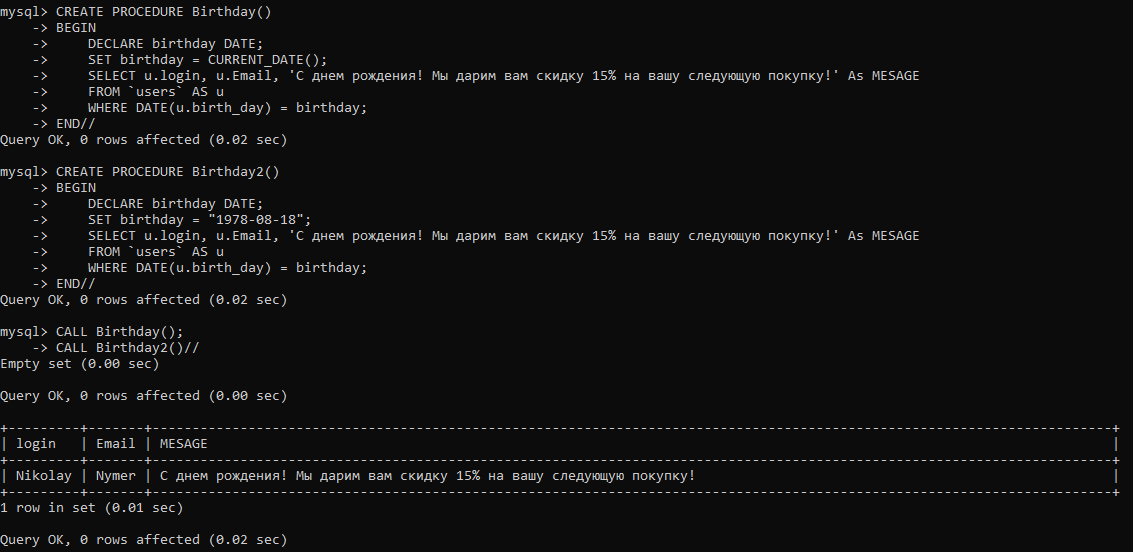
Этот код создает функцию в MySQL с именем NOK, которая принимает два целых числа (chislo1 и chislo2) и возвращает их наименьшее общее кратное (НОК).

Внутри функции используется формула для вычисления НОК на основе НОД. Возвращаемое значение функции NOK равно произведению входных чисел (chislo1 и chislo2), деленному на их наибольший общий делитель (НОД), который вычисляется с помощью вызова функции NOD(chislo1, chislo2).

Таким образом, функция NOK использует результат функции NOD для вычисления НОК двух входных чисел.

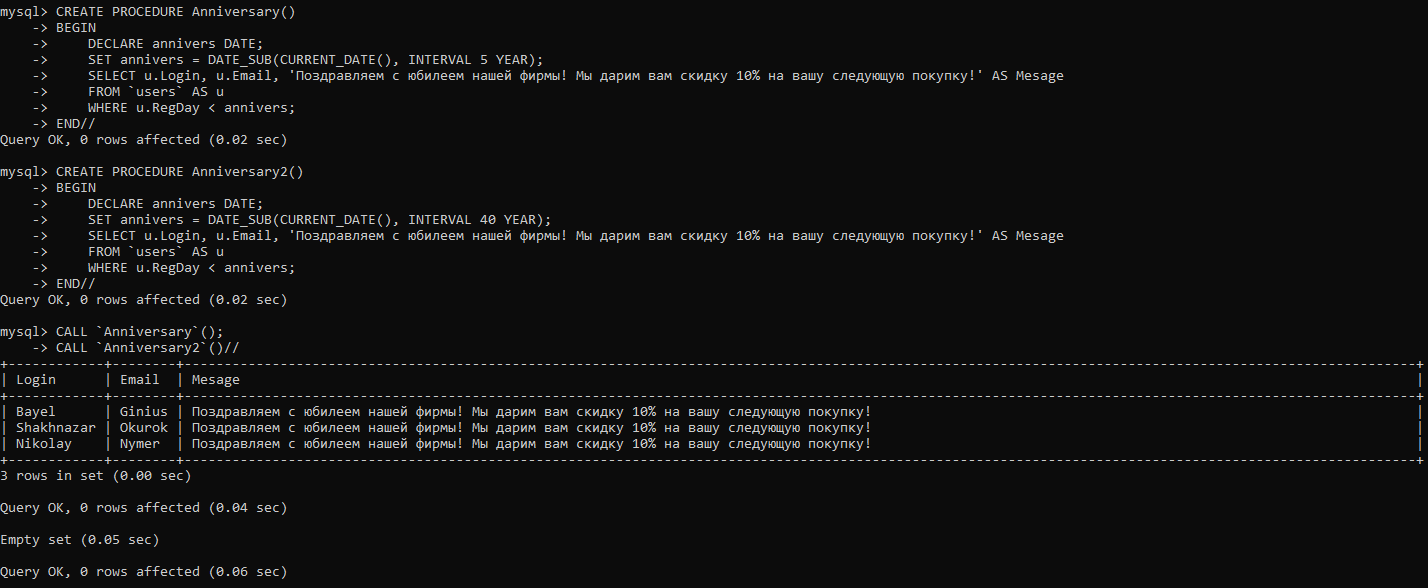
**Создать процедуры**

****

1. **Поздравить покупателей с днем рождения и предложить скидку (используйте псевдоним для столбца, чтобы проинформировать).**

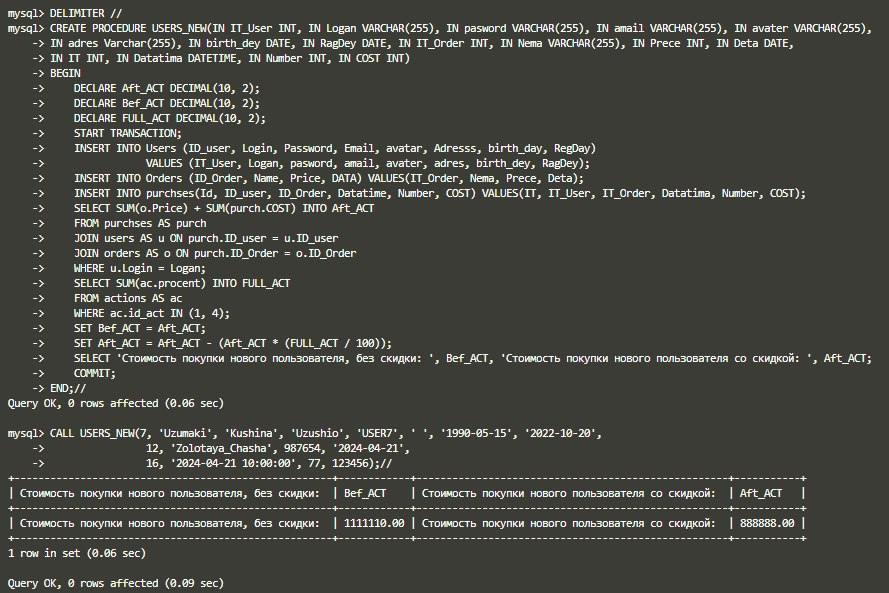
**Хранимая процедура Birthday**() выводит сообщение с поздравлением и скидкой для пользователей, у которых сегодня день рождения. Она объявляет переменную birthday типа DATE и присваивает ей значение текущей даты. Затем она выбирает из таблицы users поля login, Email и сообщение 'С днем рождения! Мы дарим вам скидку 15% на вашу следующую покупку!' для всех пользователей, у которых поле birth\_day равно переменной birthday.

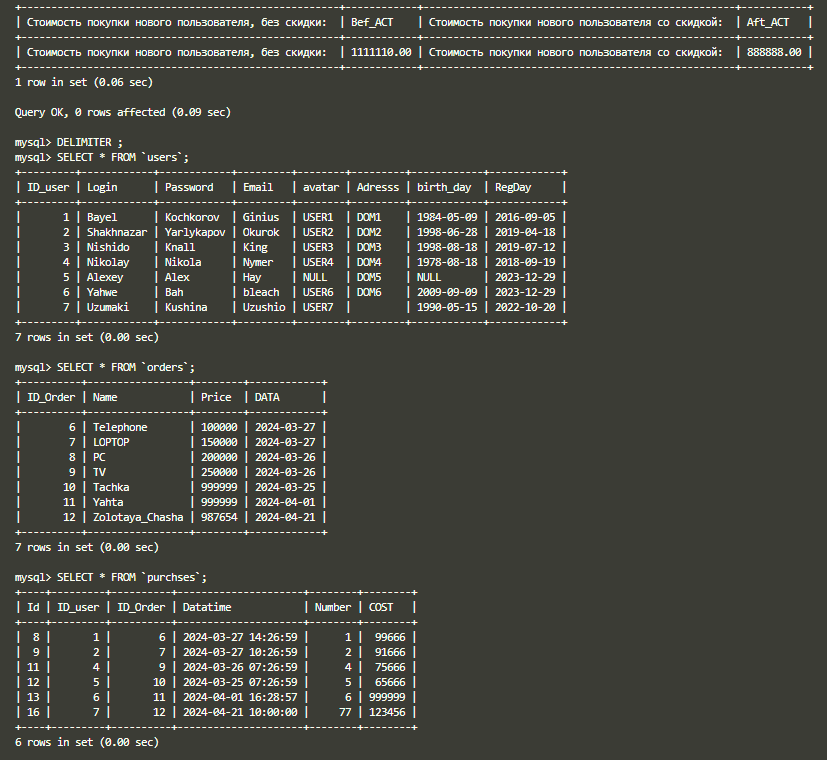
**Хранимая процедура Birthday2()** аналогична первой, но она выводит сообщение с поздравлением и скидкой для пользователей с фиксированной датой рождения (“1978-08-18”). Она объявляет переменную birthday типа DATE и присваивает ей значение “1978-08-18”. Затем она выбирает из таблицы users поля login, Email и сообщение 'С днем рождения! Мы дарим вам скидку 15% на вашу следующую покупку!' для всех пользователей, у которых поле birth\_day равно переменной birthday.

1. **В честь юбилея фирмы поздравьте и предложите скидку старым покупателям.**

Хранимая процедура Anniversary() выводит сообщение с поздравлением и скидкой для пользователей, которые зарегистрировались более 5 лет назад. Она объявляет переменную annivers типа DATE и присваивает ей значение текущей даты минус 5 лет. Затем она выбирает из таблицы users поля Login, Email и сообщение 'Поздравляем с юбилеем нашей фирмы! Мы дарим вам скидку 10% на вашу следующую покупку!' для всех пользователей, которые зарегистрировались ранее переменной annivers.

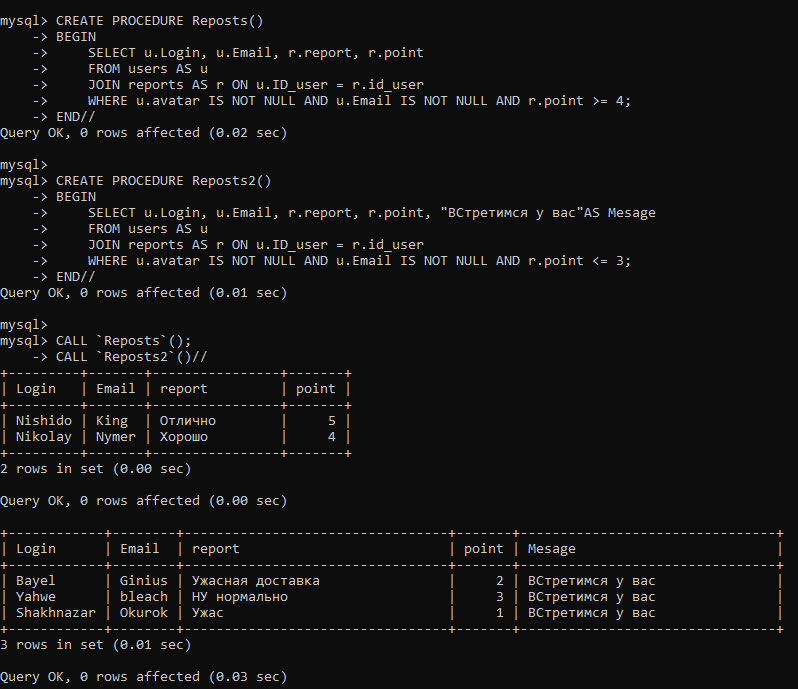
Хранимая процедура Anniversary2() аналогична первой, но она выводит сообщение с поздравлением и скидкой для пользователей, которые зарегистрировались более 40 лет назад. Она объявляет переменную annivers типа DATE и присваивает ей значение текущей даты минус 40 лет. Затем она выбирает из таблицы users поля Login, Email и сообщение 'Поздравляем с юбилеем нашей фирмы! Мы дарим вам скидку 10% на вашу следующую покупку!' для всех пользователей, которые зарегистрировались ранее переменной annivers.

1. **Добавление нового покупателя с одновременным заказом (использовать транзакции, не забыть про скидку).**

****

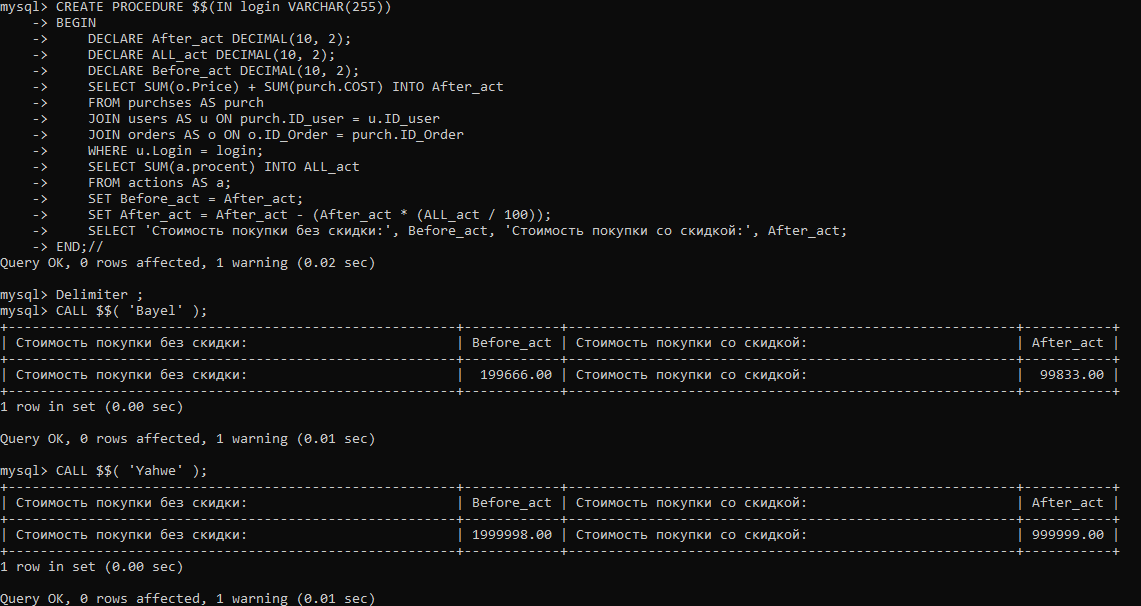
Этот код представляет собой хранимую процедуру, которая выполняет несколько действий при добавлении нового пользователя, заказа и покупки в базу данных. Процедура принимает различные параметры, такие как логин, пароль, email пользователя, данные о заказе (название, цена, дата) и данные о покупке (ID, дата-время, количество, стоимость). Внутри процедуры сначала начинается транзакция, чтобы гарантировать целостность данных. Затем происходит добавление новой записи о пользователе в таблицу Users, новой записи о заказе в таблицу Orders и новой записи о покупке в таблицу purchases. Далее процедура вычисляет общую стоимость покупок пользователя до и после применения скидки. Сумма скидки рассчитывается на основе определенных действий, которые указаны в таблице actions. Затем значения до и после применения скидки обновляются, и выводится информация о стоимости покупки до и после применения скидки. В конце процедуры транзакция завершается, чтобы сохранить все изменения в базе данных.

1. **Составьте список покупателей, имеющих аватары, e-mail и оставивших отзыв отзыв и оценку не ниже 4.**

****

**Хранимая процедура Reposts()** выбирает информацию о пользователях, которые имеют аватар и адрес электронной почты, а также написали 4 или более отчетов. Она извлекает поля Login, Email, report и point из таблиц users и reports, соединяя их по полю ID\_user.

**Хранимая процедура Reposts2()** выбирает информацию о пользователях, которые имеют аватар и адрес электронной почты, а также написали 3 или менее отчетов. Она также извлекает поля Login, Email, report и point из таблиц users и reports, соединяя их по полю ID\_user. Кроме того, она добавляет дополнительное поле Mesage со значением "ВСретимся у вас".

1. **Покупателю с заданным логином рассчитать стоимость покупки товара без скидки и со скидкой (скидки суммируются).**

Хранимая процедура $$ вычисляет стоимость покупки с учетом скидки для указанного пользователя (login). Она объявляет три переменные для хранения стоимости покупки после и до применения скидки, а также общего процента скидки. Затем она вычисляет стоимость покупки после применения скидки путем суммирования стоимости всех заказов и покупок для указанного пользователя. Далее вычисляется общий процент скидки как сумма процентов скидки по всем действиям. Стоимость покупки до применения скидки вычисляется как значение стоимости покупки после применения скидки. Наконец, к стоимости покупки применяется скидка, и на экран выводится информация о стоимости покупки до и после применения скидки.