# **Практическое задание по теме «Операторы, фильтрация, сортировка и ограничение»**

1. *Пусть в таблице users поля created\_at и updated\_at оказались незаполненными. Заполните их текущими датой и временем.*

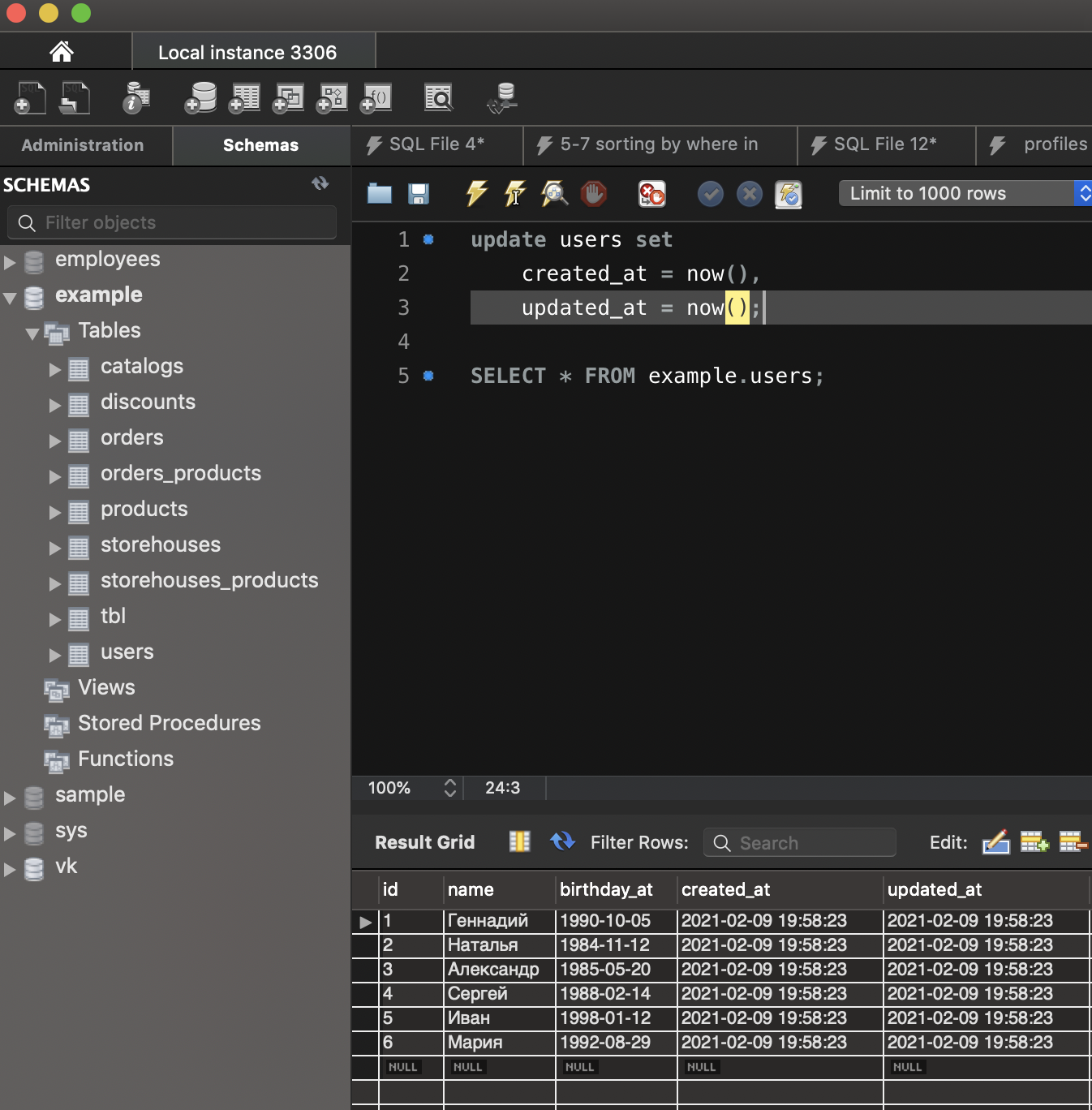
**Решение:**

update users set

created\_at = now(),

updated\_at = now();

**Результат:**



1. *Таблица users была неудачно спроектирована. Записи created\_at и updated\_at были заданы типом VARCHAR и в них долгое время помещались значения в формате 20.10.2017 8:10. Необходимо преобразовать поля к типу DATETIME, сохранив введённые ранее значения.*

**Решение:**

1. Подготовка таблицы к заданию

ALTER TABLE `example`.`users`

CHANGE COLUMN `created\_at` `created\_at` VARCHAR(45) NULL DEFAULT '20.10.2017 8:10' ,

CHANGE COLUMN `updated\_at` `updated\_at` VARCHAR(45) NULL DEFAULT '20.10.2017 8:10' ;

1. Изменяем данные в таблице из строки в формат даты

update example.users set

updated\_at = str\_to\_date(updated\_at, "%d.%m.%Y %h:%i"),

created\_at = str\_to\_date(created\_at, "%d.%m.%Y %h:%i");

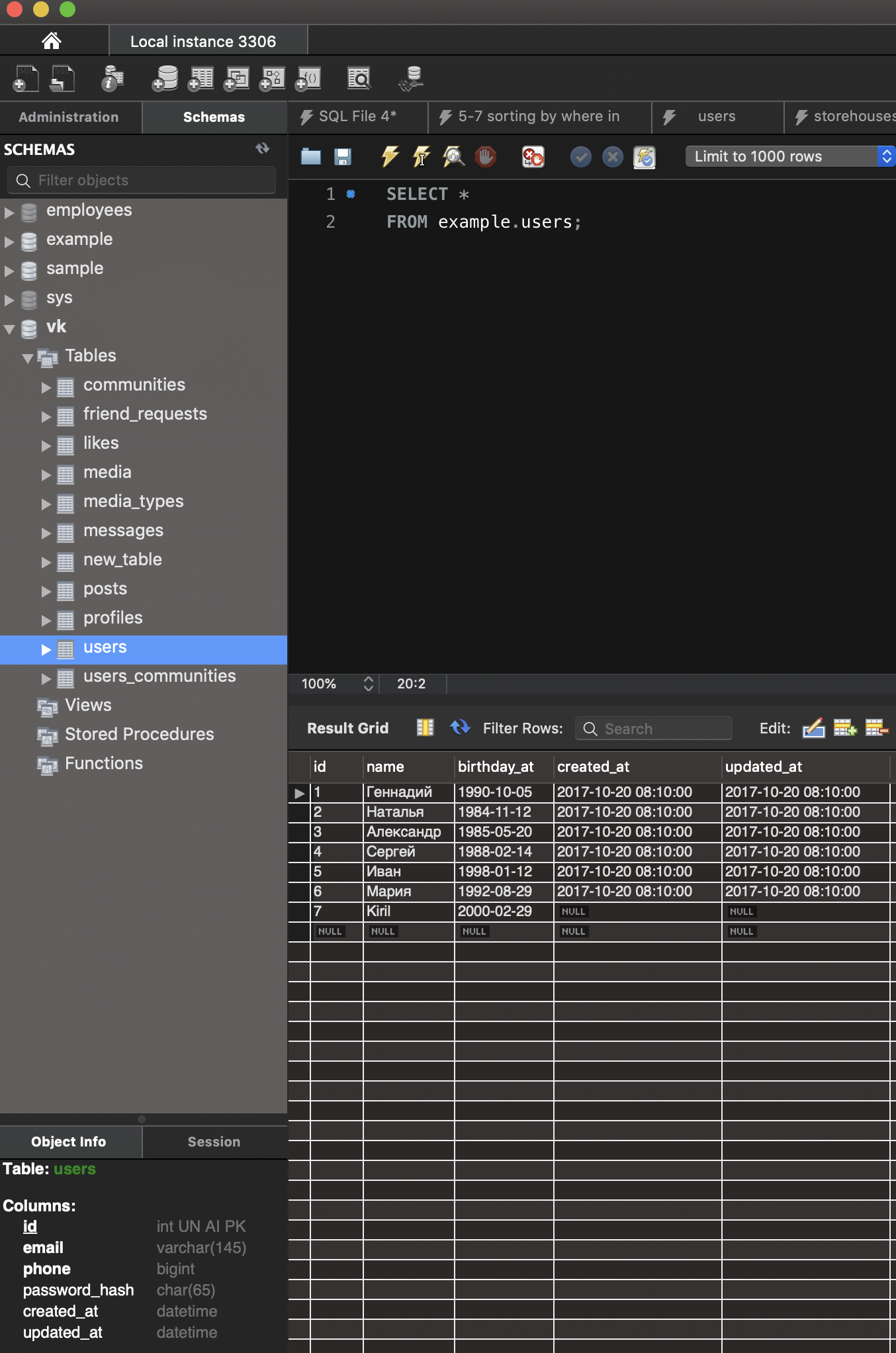
1. Затем меняем тип данных у колонок

ALTER TABLE `example`.`users`

CHANGE COLUMN `created\_at` `created\_at` DATETIME NULL DEFAULT NULL ,

CHANGE COLUMN `updated\_at` `updated\_at` DATETIME NULL DEFAULT NULL ;

**Результат:**

****

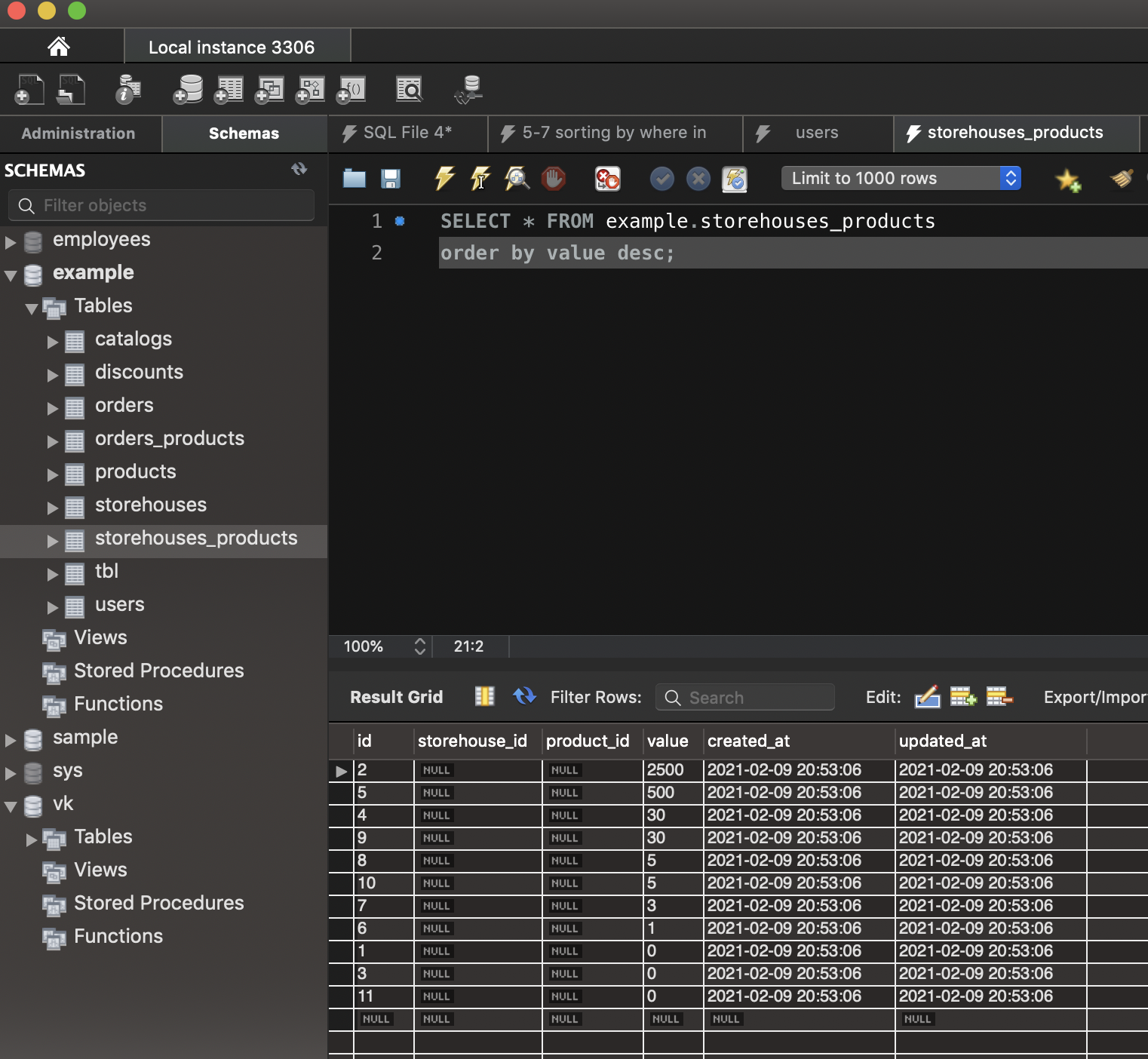
1. *В таблице складских запасов storehouses\_products в поле value могут встречаться самые разные цифры: 0, если товар закончился и выше нуля, если на складе имеются запасы. Необходимо отсортировать записи таким образом, чтобы они выводились в порядке увеличения значения value. Однако нулевые запасы должны выводиться в конце, после всех записей.*

**Решение:**

SELECT \* FROM example.storehouses\_products

order by value desc;

**Результат:**



1. *(по желанию) Из таблицы users необходимо извлечь пользователей, родившихся в августе и мае. Месяцы заданы в виде списка английских названий (may, august)*

**Решение:**

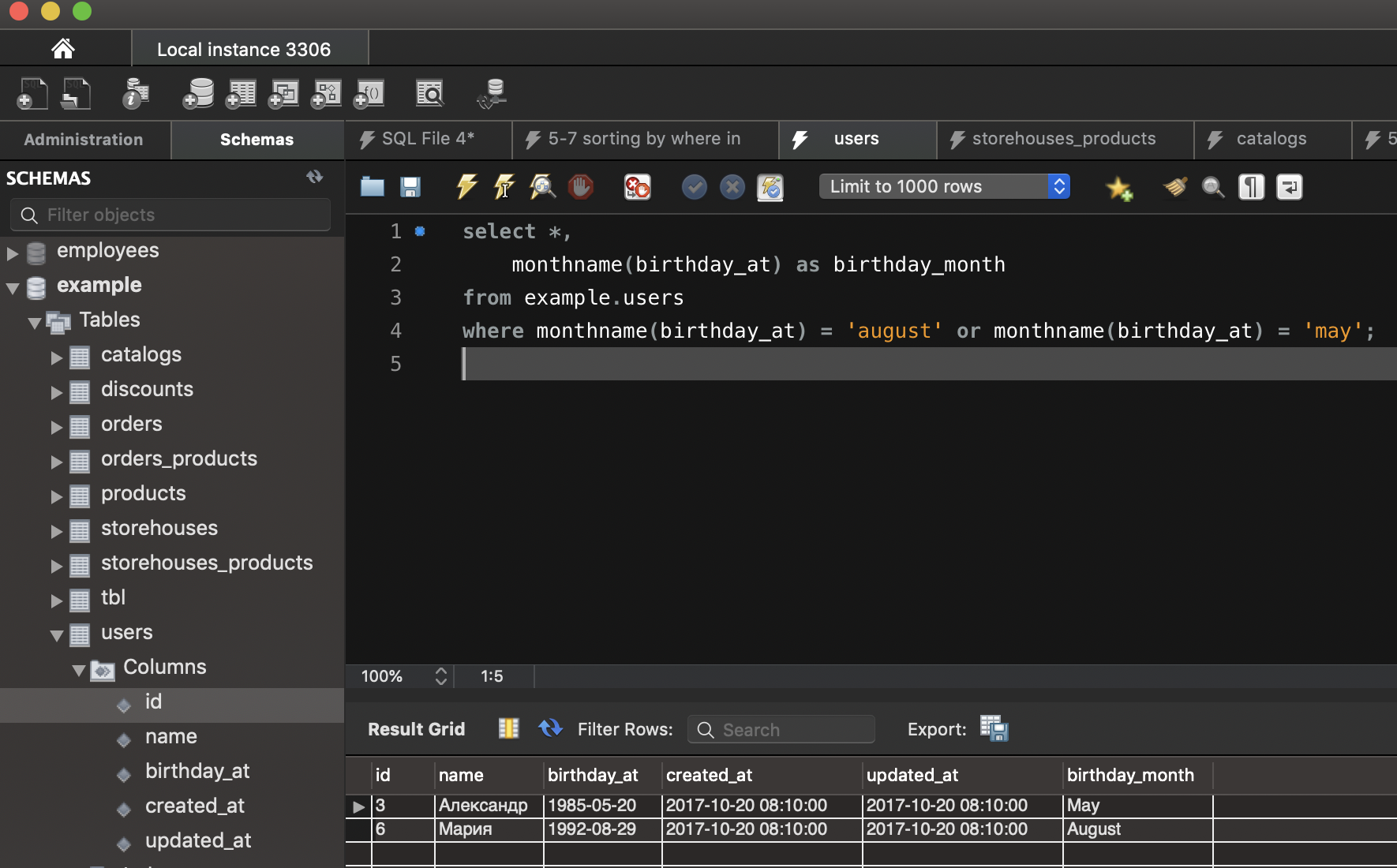
select \*,

monthname(birthday\_at) as birthday\_month

from example.users

where monthname(birthday\_at) = 'august' or monthname(birthday\_at) = 'may';

**Результат:**



1. *(по желанию) Из таблицы catalogs извлекаются записи при помощи запроса. SELECT \* FROM catalogs WHERE id IN (5, 1, 2); Отсортируйте записи в порядке, заданном в списке IN.*

**Решение:**

SELECT \* FROM example.catalogs

where id in (5, 1, 2)

order by(

case id

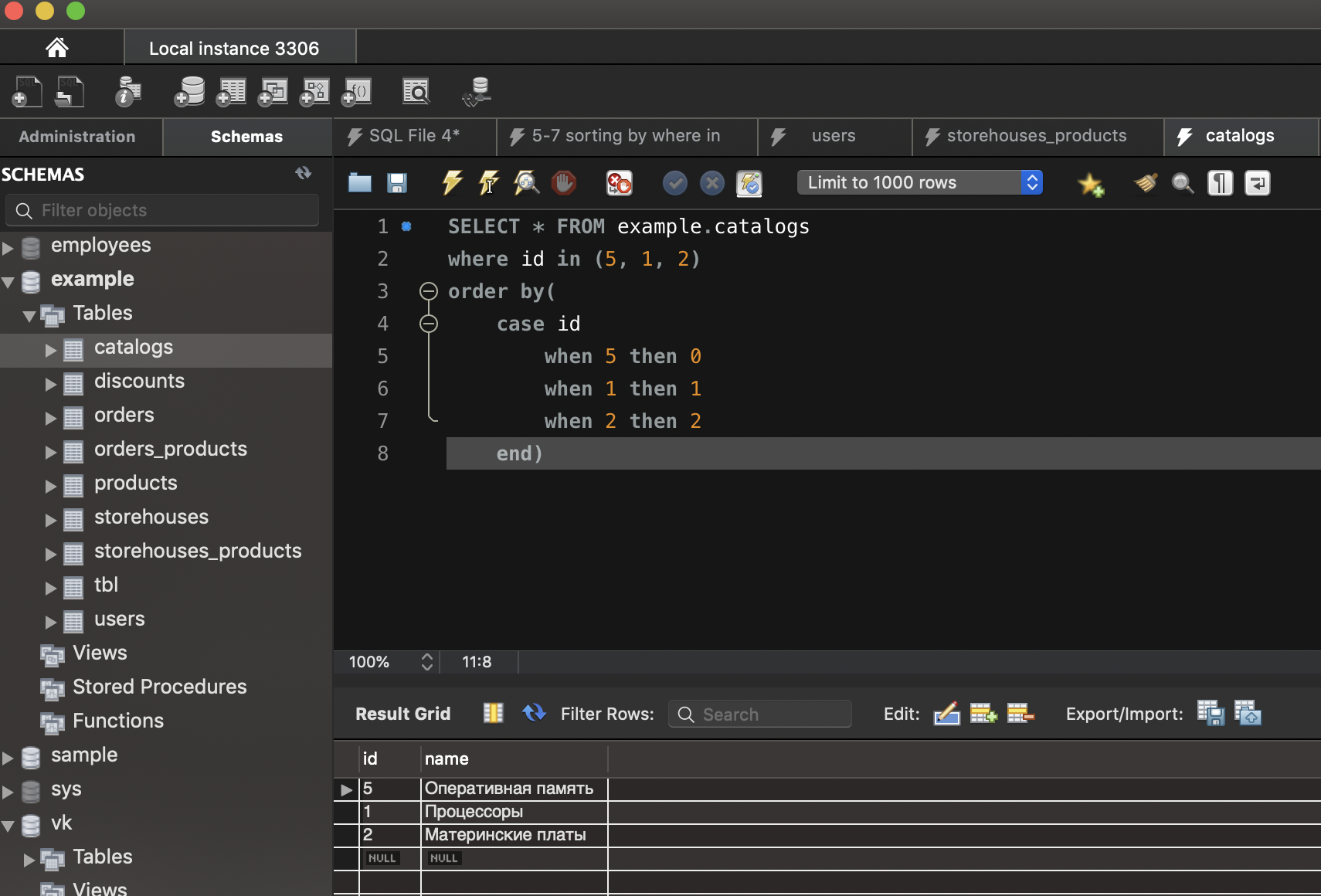
when 5 then 0

when 1 then 1

when 2 then 2

end);

**Результат:**



# **Практическое задание теме «Агрегация данных»**

1. *Подсчитайте средний возраст пользователей в таблице users.*

**Решение:**

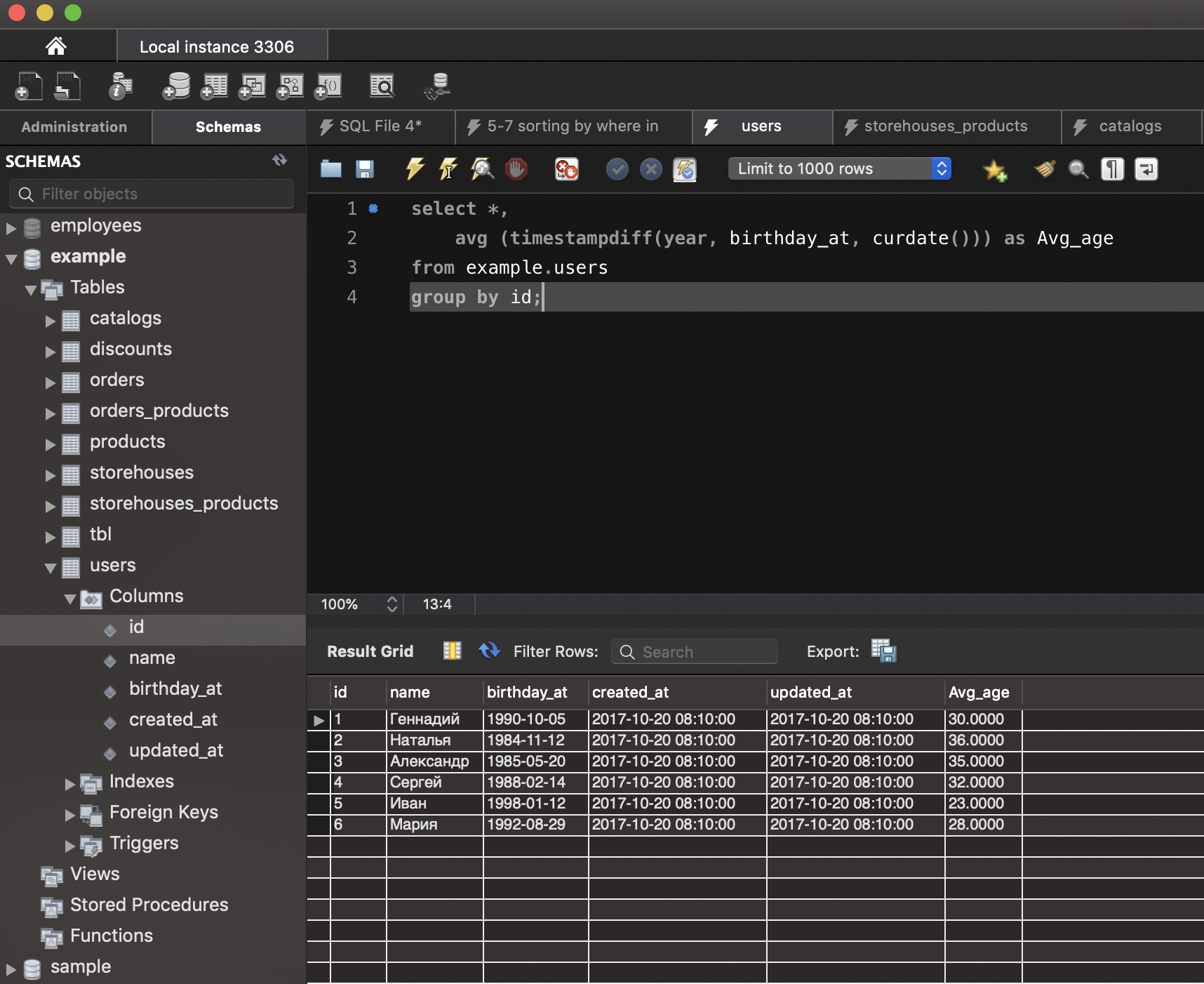
select \*,

avg (timestampdiff(year, birthday\_at, curdate())) as Avg\_age

from example.users

group by id;

**Результат:**



1. *Подсчитайте количество дней рождения, которые приходятся на каждый из дней недели. Следует учесть, что необходимы дни недели текущего года, а не года рождения.*

**Решение:**

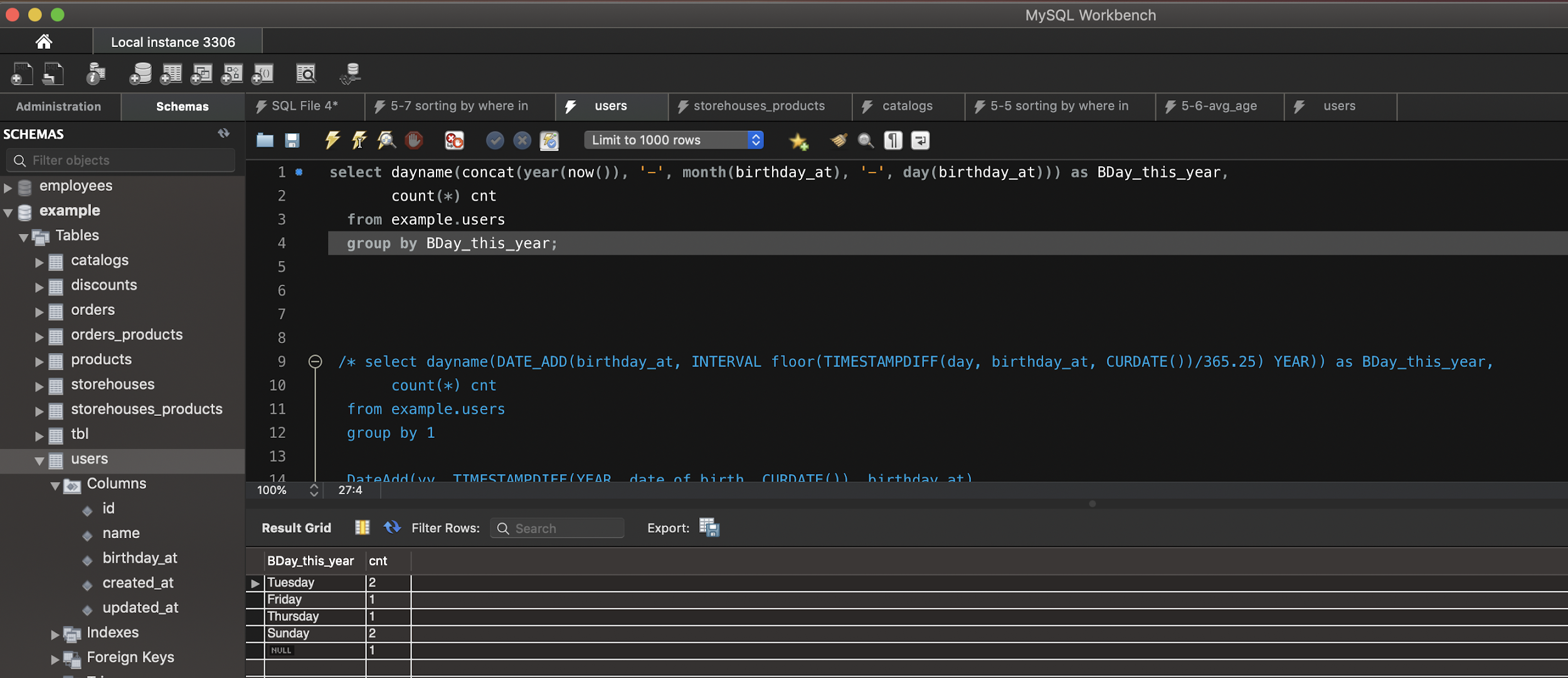
select dayname(concat(year(now()), '-', month(birthday\_at), '-', day(birthday\_at))) as BDay\_this\_year,

count(\*) cnt

from example.users

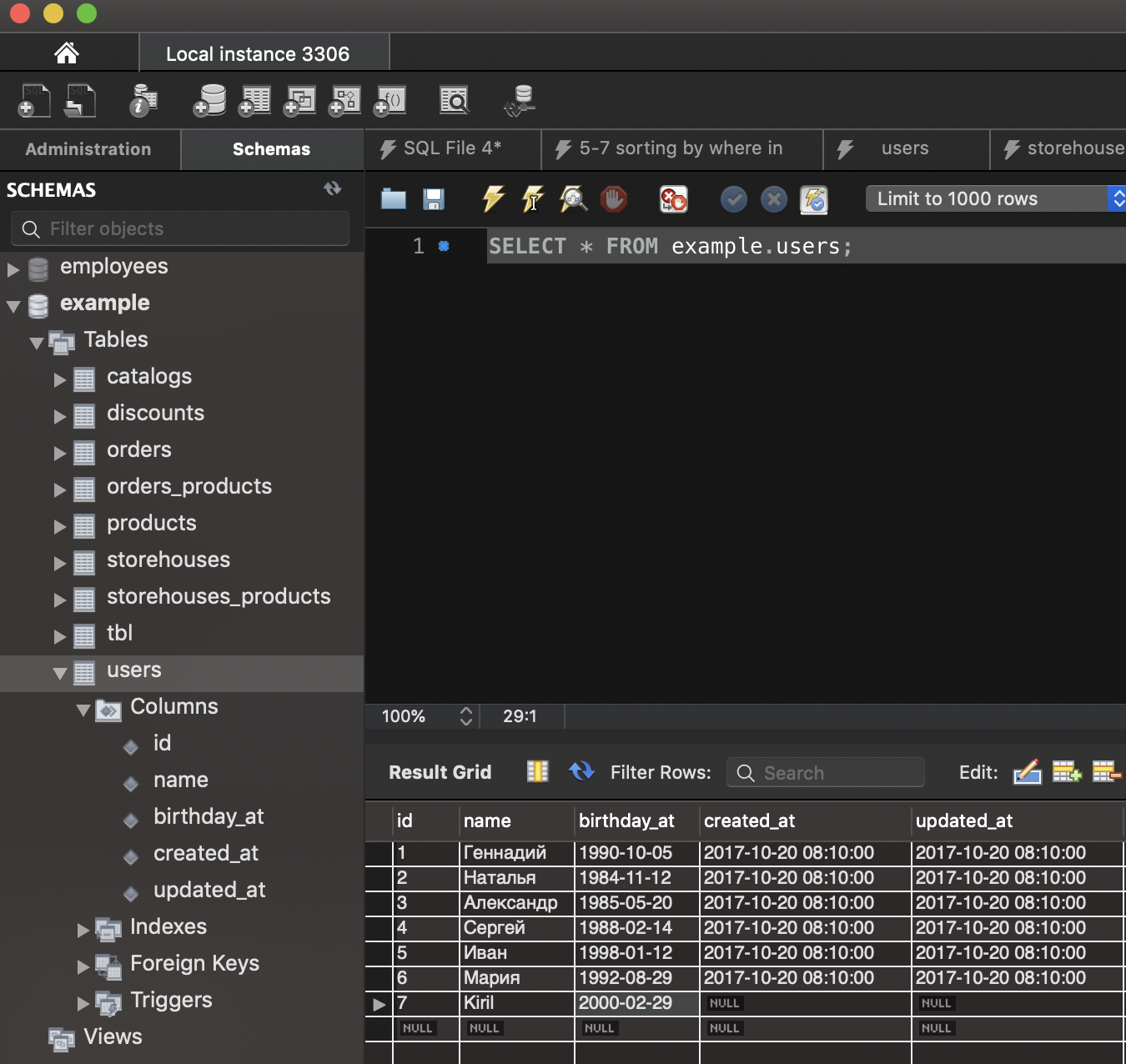
group by BDay\_this\_year

**Результат:**



Вариант решения выше не учитывает високосный год, судя по всему ошибка не выдается, а выдается null.

Для теста я добавила еще одного пользователя Kiril с днем рождения в високосный год и попробовала решить его по другому ниже:



**Решение:**

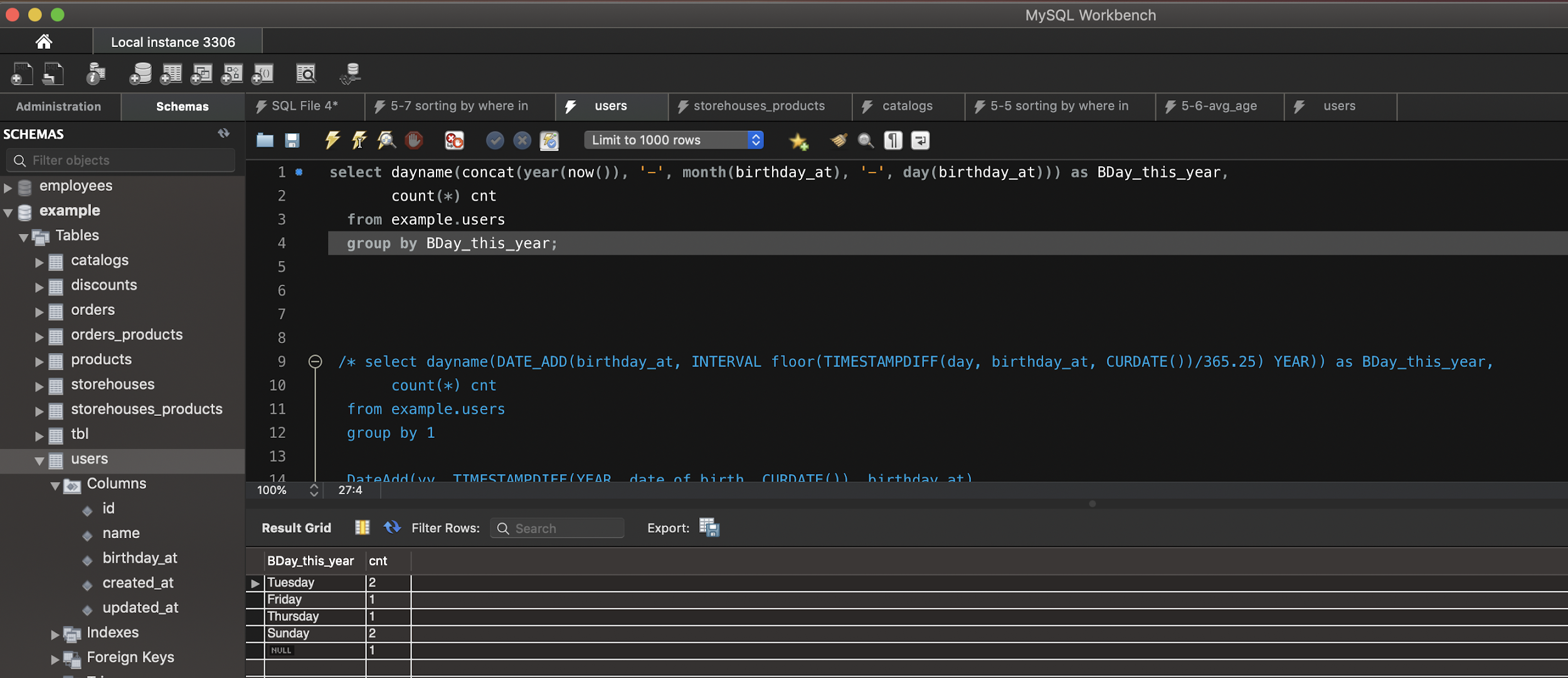
select dayname(DATE\_ADD(birthday\_at, INTERVAL floor(TIMESTAMPDIFF(day, birthday\_at, CURDATE())/365.25) YEAR)) as BDay\_this\_year,

count(\*) cnt

from example.users

group by BDay\_this\_year

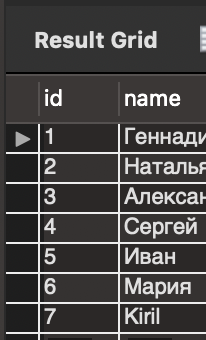
**Результат:**



1. *(по желанию) Подсчитайте произведение чисел в столбце таблицы.*

**Решение:**

Произведение производится на колонку id



SELECT

round(exp(sum(ln(id))))

FROM example.users;

**Результат:**

