**7. ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ**

**7.1 МАТЕМАТИЧЕСКИЕ ФУНКЦИИ**

**Задача 7.1.1 Критическая температура**

В таблице **experiments** содержится информация о проведенных экспериментах в институте физики. Для каждого эксперимента фиксировалась дата и температура.

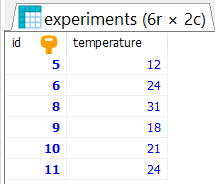
Нормальная температура колеблется в пределах от -10 до +10 градусов включительно. Если температура выходит за эти пределы в любую из сторон, то она считается **критической**.

Выведите эксперименты с критической температурой.  
В итоговой таблице нужно оставить только номер эксперимента (id) и абсолютное значение температуры (temperature).

Таблица experiments

|  |  |  |
| --- | --- | --- |
| **id** | **temperature** | **date** |
| 1 | 3 | 2017-03-09 |
| 2 | -8 | 2017-03-09 |
| 3 | 4 | 2017-03-10 |
| 4 | 4 | 2017-03-10 |
| 5 | 12 | 2017-03-10 |
| 6 | 24 | 2017-03-11 |
| 7 | -8 | 2017-03-11 |
| 8 | 31 | 2017-03-13 |
| 9 | -18 | 2017-03-14 |
| 10 | -21 | 2017-03-14 |
| 11 | 24 | 2017-03-16 |
| 12 | 3 | 2017-03-16 |
| 13 | 5 | 2017-03-17 |
| 14 | -6 | 2017-03-18 |

**select** id, **ABS**(temperature) **AS** temperature **from** experiments **where** temperature < -10 **OR** temperature > 10;



**Задача 7.1.2 ТОП 5 популярных фильмов**

В таблице **films** содержится информация о фильмах: id, рейтинг, год выпуска и количество голосов. Получите название фильма, а также его рейтинг округленный до 2 знаков после запятой (например 4.34).

Отсортируйте результаты в порядке убывания рейтинга и оставьте только 5 записей.

Поле с округленным рейтингом должно называться **rating**.

Таблица films

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **name** | **rating** | **year** | **votes** |
| 1 | Зеленая миля | 9.135 | 1999 | 431238 |
| 2 | Бойцовский клуб | 8.715 | 1999 | 410297 |
| 3 | Форрест Гамп | 9.013 | 1994 | 421514 |
| 4 | Побег из Шоушенка | 9.191 | 1994 | 460078 |
| 5 | Список Шиндлера | 8.884 | 1993 | 233033 |
| 6 | Иван Васильевич меняет профессию | 8.707 | 1973 | 320339 |
| 7 | Леон | 8.778 | 1994 | 351308 |
| 8 | Начало | 8.775 | 2010 | 484028 |
| 9 | 1+1 | 8.838 | 2011 | 444942 |
| 10 | Король Лев | 8.758 | 1994 | 300302 |

**select** **name**, **round**(rating,2) **as** rating **from** films **order** **by** rating **desc** **limit** 5



**Задача 7.1.3 Округление в сторону покупателя**

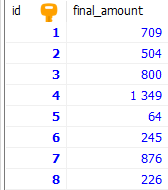
Для удобства покупателей, в продуктовом магазине решили округлять суммы в чеках в сторону покупателей. Если стоимость товаров получилась в 250 рублей 74 копейки, то в чеке будет 250 рублей ровно.

Получите из таблицы **bills** id чека и также конечную сумму в чеке.  
Колонку с конечной ценой назовите **final\_amount**.

При расчете учитывайте персональную скидку покупателя. Cперва нужно уменьшить базовую цену (**amount**) на величину скидки, а после вычислить конечную сумму (**final\_amount**). Величина скидки в процентах.

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **user\_id** | **amount** | **personal\_sale** |
| 1 | 745 | 738.65 | 4 |
| 2 | 1445 | 531.12 | 5 |
| 3 | NULL | 800.0 | 0 |
| 4 | NULL | 1349.0 | 0 |
| 5 | 56137 | 67.5 | 4 |
| 6 | 171 | 245.14 | 0 |
| 7 | NULL | 876.59 | 0 |
| 8 | 6735 | 234.0 | 3 |

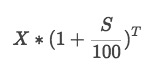
**select** id, **floor**((1-personal\_sale/100)\*amount) **as** final\_amount **FROM** bills



**Задача 7.1.4 Капитализация процентов**

Во многих видах банковских вкладов есть **капитализация процентов** — причисление процентов к сумме вклада, что позволяет в дальнейшем осуществлять начисление процентов на проценты.

Общая сумма, которую получит вкладчик, с учетом капитализации считается по следующей формуле:



где X – сумма начального вклада (**initial\_sum**),  
S – величина процентов (**percent**),  
T – срок вклада в годах (**years**).

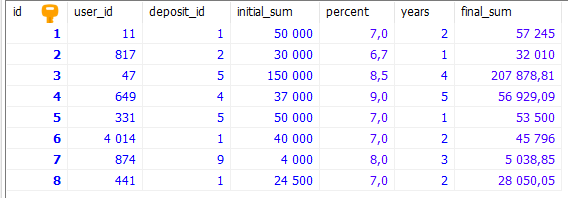
Посчитайте суммы, которые получат вкладчики по истечении сроков вкладов.  
Выведите сперва все столбцы таблицы, а в конце добавьте колонку **final\_sum** с результатом вычислений.

Данные в **final\_sum** округлите до 2 знаков после точки.

Таблица deposits

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **id** | **user\_id** | **deposit\_id** | **initial\_sum** | **percent** | **years** |
| 1 | 11 | 1 | 50000 | 7 | 2 |
| 2 | 817 | 2 | 30000 | 6.7 | 1 |
| 3 | 47 | 5 | 150000 | 8.5 | 4 |
| 4 | 649 | 4 | 37000 | 9 | 5 |
| 5 | 331 | 5 | 50000 | 7 | 1 |
| 6 | 4014 | 1 | 40000 | 7 | 2 |
| 7 | 874 | 9 | 4000 | 8 | 3 |
| 8 | 441 | 1 | 24500 | 7 | 2 |

**select** id, user\_id, deposit\_id, initial\_sum, percent, years, **round**((initial\_sum\***power**((1+percent/100),years)),2) **as** final\_sum **FROM** deposits

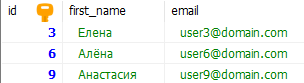


**Задача 7.1.5 Каждый третий**

В компании захотели провести e-mail рассылку с акционным предложением. Но перед отправкой всех писем решили сделать тестовый запуск на одной трети аудитории. И для равномерного распределения пользователей решили отправлять письма каждому третьему клиенту (id равны 3, 6, 9 и тд).Выберите из таблицы **users** каждого третьего пользователя. Выведите только id, имя и email. Таблица users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **first\_name** | **last\_name** | **email** | **age** |
| 1 | Виктор | Алтушев | user1@domain.com | 20 |
| 2 | Светлана | Иванова | user2@domain.com | 17 |
| 3 | Елена | Абрамова | user3@domain.com | 18 |
| 4 | Василиса | Кац | user4@domain.com | 15 |
| 5 | Антон | Сорокин | user5@domain.com | 22 |
| 6 | Алёна | Алясева | user6@domain.com | 28 |
| 7 | Лиана | Белая | user7@domain.com | 21 |
| 8 | Карина | Белая | user8@domain.com | 30 |
| 9 | Анастасия | Дейчман | user9@domain.com | 16 |
| 10 | Юлия | Фёдорова | user10@domain.com | 25 |

**select** id, first\_name, email **from** users **where** id%3=0

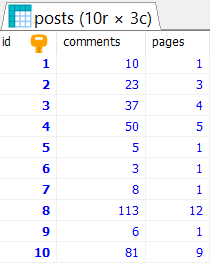


**Задача 7.1.6 Страницы**

Комментарии к посту на сайте разбиты на страницы – по 10 комментариев на каждой. Выведите из таблицы **posts** id поста, количество комментариев (**comments**) и количество страниц (**pages**), которое необходимо для вывода этих комментариев.Таблица posts

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **title** | **author\_id** | **comments** |
| 1 | Массовая атака на оборудование Cisco | 43 | 10 |
| 2 | Знакомство с новым элементом dialog | 218 | 23 |
| 3 | Что такое «Большие данные» | 7361 | 37 |
| 4 | Электронные книги и их форматы | 7 | 50 |
| 5 | Мангровый лес: крутейший биом планеты | 245 | 5 |
| 6 | Электроотвёртка своими руками или наш ответ Xiaomi | 144 | 3 |
| 7 | Реверс инжиниринг процессора ARM1 | 116 | 8 |
| 8 | Программирование современных микроконтроллеров | 565 | 113 |
| 9 | Переработка мусора: как уменьшить количество свалок | 4515 | 6 |
| 10 | Подключаем счетчик воды к умному дому | 33 | 81 |

**select** id, comments, **ceiling**(comments/10) **as** pages **from** posts

****

**7.2 СТРОКОВЫЕ ФУНКЦИИ**

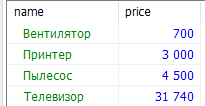
**Задача 7.2.1 Длина имени**

Получите из таблицы **products** имена и цены всех товаров, у которых длина имени от 5 до 10 символов включительно.

Отсортируйте данные по имени в алфавитном порядке. Таблица products

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **name** | **count** | **price** |
| 1 | Стиральная машина | 5 | 10000 |
| 2 | Холодильник | 0 | 10000 |
| 3 | Микроволновка | 3 | 4000 |
| 4 | Пылесос | 2 | 4500 |
| 5 | Вентилятор | 0 | 700 |
| 6 | Телевизор | 7 | 31740 |
| 7 | Утюг | 2 | 2500 |
| 8 | Принтер | 4 | 3000 |

**select** **name**, price **from** products **where** **char\_length**(**name**) >=5 **and** **char\_length**(**name**) <=10 **order** **by** **name**



**Задача 7.2.2 Паспорта**

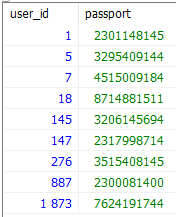
В таблице **passports** хранятся данные о паспортах пользователей. Серия и номер паспорта хранятся в отдельных строковых столбцах.

Получите из таблицы id пользователя и его паспорт в формате **SSSSNNNNNN**, где **SSSS** — серия, а **NNNNNN** — номер).  
Итоговый номер паспорта должен иметь название **passport**.

Данные отсортируйте по id пользователя.Таблица passports

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **user\_id** | **series** | **number** |
| 1 | 1 | 2301 | 148145 |
| 2 | 5 | 3295 | 409144 |
| 3 | 145 | 3206 | 145694 |
| 4 | 147 | 2317 | 998714 |
| 5 | 276 | 3515 | 408145 |
| 6 | 18 | 8714 | 881511 |
| 7 | 7 | 4515 | 009184 |
| 8 | 1873 | 7624 | 191744 |
| 9 | 887 | 2300 | 081400 |

**select** user\_id, **concat**(series, **number**) **as** passport **from** passports **order** **by** user\_id



**Задача 7.2.3 Номер паспорта**

В таблице **passports** содержатся данные о паспортах пользователей. Серия и номер паспорта хранятся в отдельных целочисленных столбцах, что облегчает сортировку, но затрудняет вывод данных.

Например, если мы выполним запрос:

SELECT user\_id, CONCAT\_WS(" ", series, number) as passport FROM passports;

То для серии равной «21» и номера равному «81400» получим «21 81400».

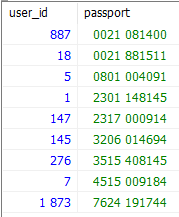
Однако при выводе длина серии должна составлять 4 символа, а длина номера — 6 символов.

Измените исходный запрос так, чтобы короткие серии и номера получили лидирующие нули: вместо «21 81400» должно выводиться **«0021 081400»**.

Данные отсортируйте сперва по серии, а потом по номеру. Таблица passports

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **user\_id** | **series** | **number** |
| 1 | 1 | 2301 | 148145 |
| 2 | 5 | 801 | 4091 |
| 3 | 145 | 3206 | 14694 |
| 4 | 147 | 2317 | 914 |
| 5 | 276 | 3515 | 408145 |
| 6 | 18 | 21 | 881511 |
| 7 | 7 | 4515 | 9184 |
| 8 | 1873 | 7624 | 191744 |
| 9 | 887 | 21 | 81400 |

**SELECT** user\_id, **CONCAT\_WS**(" ", **lpad**(series,4,0), **lpad**(**number**,6,0)) **as** passport **FROM** passports **order** **by** series, **number**;



**Задача 7.2.4 Фамилия, имя, отчество**

Выведите из таблицы **users** два столбца.

В первый поместите **id** пользователя, а во второй его **Фамилию Имя Отчество** разделенные пробелами. Второй столбец назовите — **name**.

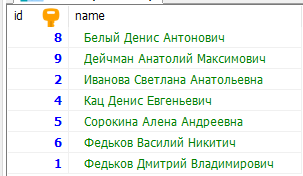
Данные должны быть отсортированы в алфавитном порядке сперва по фамилии, затем по имени и отчеству.

Выводить нужно только тех пользователей, у которых есть отчество.

Таблица users

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **first\_name** | **last\_name** | **patronymic** |
| 1 | Дмитрий | Федьков | Владимирович |
| 2 | Светлана | Иванова | Анатольевна |
| 3 | Никита | Трутнев |  |
| 4 | Денис | Кац | Евгеньевич |
| 5 | Алена | Сорокина | Андреевна |
| 6 | Василий | Федьков | Никитич |
| 7 | Яровслав | Сорокин |  |
| 8 | Денис | Белый | Антонович |
| 9 | Анатолий | Дейчман | Макcимович |
| 10 | Юлия | Фёдорова |  |

**SELECT** id, **CONCAT\_WS**(" ", last\_name, first\_name, patronymic) **as** **name** **FROM** users **where** patronymic != '' **order** **by** last\_name, first\_name, patronymic



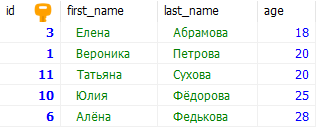
**Задача 7.2.5 Фамилии на ...ова**

Получите из таблицы **users** всех пользователей от 18 лет и старше с фамилией, заканчивающейся на «ова». Отсортируйте результаты по возрасту, а затем по фамилии.

PS. В итоговой таблице нужно выводить все столбцы. Таблица users

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **first\_name** | **last\_name** | **age** |
| 1 | Вероника | Петрова | 20 |
| 2 | Светлана | Иванова | 17 |
| 3 | Елена | Абрамова | 18 |
| 4 | Василиса | Кац | 15 |
| 5 | Алёна | Сорокина | 22 |
| 6 | Алёна | Федькова | 28 |
| 7 | Лиана | Сорокина | 21 |
| 8 | Карина | Белая | 30 |
| 9 | Анастасия | Дейчман | 16 |
| 10 | Юлия | Фёдорова | 25 |
| 11 | Татьяна | Сухова | 20 |

**SELECT** \* **FROM** users **where** last\_name **like** '%ова' **and** age >= 18 **order** **by** age, last\_name



**Задача 7.2.6 Номер и серия паспорта**

В таблице **users** в поле **passport** хранятся данные о паспорте пользователя в символьном формате. В одном поле хранится и серия (первые 4 цифры) и номер, последние 6 цифр. Если у пользователя нет паспорта, то ничего не содержит — NULL.

Получите **id** пользователя, а также раздельно серию паспорта и его номер. Серия должна быть в колонке **series**, а номер в колонке **number**.

Пользователей без паспорта исключите.

Таблица users

|  |  |  |
| --- | --- | --- |
| **id** | **email** | **passport** |
| 1 | user1@domain.com | 2301148145 |
| 2 | user2@domain.com | 3295409144 |
| 3 | user3@domain.com | 3206145694 |
| 4 | user4@domain.com | NULL |
| 5 | user5@domain.com | 3515408145 |
| 6 | user6@domain.com | 8714881511 |
| 7 | user7@domain.com | 4515009184 |
| 8 | user8@domain.com | NULL |
| 9 | user9@domain.com | 2300081400 |

**SELECT** id, **left**(passport, 4) **as** series, **right**(passport, 6) **as** **number** **FROM** users **where** passport >0



**Задача 7.2.7 Убираем пробелы**

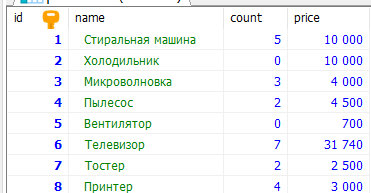
В таблице **products** хранятся данные о товарах в компании. Во время внесения данных, менеджеры часто вставляли пробелы до и после названий. Обновите имена продуктов так, чтобы в начале и в конце имен не было пробелов. Таблица products

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **name** | **count** | **price** |
| 1 | Стиральная машина | 5 | 10000 |
| 2 | Холодильник | 0 | 10000 |
| 3 | Микроволновка | 3 | 4000 |
| 4 | Пылесос | 2 | 4500 |
| 5 | Вентилятор | 0 | 700 |
| 6 | Телевизор | 7 | 31740 |
| 7 | Тостер | 2 | 2500 |
| 8 | Принтер | 4 | 3000 |

**UPDATE** products

**SET** **name** = **trim**(**name**);

**SELECT** \* **FROM** products



**Задача 7.2.8 Доменные имена**

В таблице **domains** содержится список доменных имен. Некоторые домены оканчиваются на точку, например **(google.com.)**. Обновите таблицу так, чтобы доменные имена не заканчивались точкой.

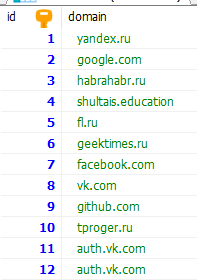
Таблица domains

|  |  |
| --- | --- |
| **id** | **domain** |
| 1 | yandex.ru |
| 2 | google.com. |
| 3 | habrahabr.ru |
| 4 | shultais.education. |
| 5 | fl.ru |
| 6 | geektimes.ru |
| 7 | facebook.com. |
| 8 | vk.com |
| 9 | github.com. |
| 10 | tproger.ru |
| 11 | auth.vk.com |
| 12 | auth.vk.com. |

**UPDATE** domains

**SET** domain = **trim**(**trailing** '.' **from** domain);

**SELECT** \* **FROM** domains



**Задача 7.2.9 Меняем таблицу**

В таблице **users** в поле **name** хранятся имя и фамилия пользователя разделенные пробелами.

1. Создайте в таблице два дополнительных поля: **first\_name** и **last\_name** длиной 50 символов.

Оба поля по умолчанию должны содержать пустую строку.

2. Заполните **first\_name** именами пользователя, а **last\_name** — фамилиями. Используйте данные из общего поля **name**.

3. В конце удалите поле **name**.

Таблица users

|  |  |  |
| --- | --- | --- |
| **id** | **name** | **age** |
| 1 | Вероника Петрова | 20 |
| 2 | Светлана Иванова | 17 |
| 3 | Елена Абрамова | 18 |
| 4 | Василиса Кац | 15 |
| 5 | Алёна Сорокина | 22 |
| 6 | Алёна Федькова | 28 |
| 7 | Лиана Сорокин | 21 |
| 8 | Карина Белая | 30 |
| 9 | Анастасия Дейчман | 16 |
| 10 | Юлия Фёдорова | 25 |

**alter** **table** users **add** **column** first\_name **varchar**(50) **default** **NULL**;

**alter** **table** users **add** **column** last\_name **varchar**(50) **default** **NULL**;

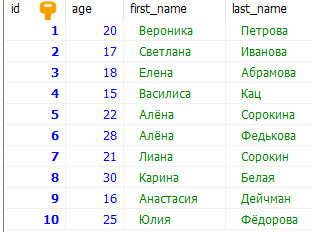
**UPDATE** users

**SET** first\_name = **SUBSTRING\_INDEX**(**name**, ' ', 1),

last\_name = **SUBSTRING\_INDEX**(**name**, ' ', -1);

**alter** **table** users **drop** **column** **name**;

**SELECT** \* **FROM** users



**7.3 ФУНКЦИИ ДАТЫ**

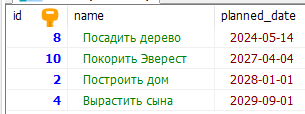
**Задача 7.3.1 Будущие задачи**

Получите из таблицы **tasks** все задачи, запланированные на будущее. Данные отсортируйте в хронологическом порядке по полю **planned\_date**.

Таблица tasks

|  |  |  |
| --- | --- | --- |
| **id** | **name** | **planned\_date** |
| 1 | Прибраться | 2017-12-31 |
| 2 | Построить дом | 2027-01-01 |
| 3 | Помыть посуду | 2018-01-03 |
| 4 | Вырастить сына | 2028-09-01 |
| 5 | Купить продукты | 2017-12-30 |
| 6 | Починить кран | 2018-01-05 |
| 7 | Помыть машину | 2018-03-03 |
| 8 | Посадить дерево | 2024-05-14 |
| 9 | Помыть окна | 2018-01-07 |
| 10 | Покорить Эверест | 2027-04-04 |

**select** \* **from** tasks **where** planned\_date>**now**() **order** **by** planned\_date

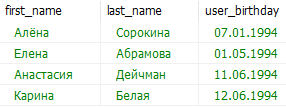


**Задача 7.3.2 Рожденные в 1994**

Получите из таблицы **users** имена, фамилии и даты рождений пользователей, которые родились в 1994 году. Дату следует вывести в формате **ДД.ММ.ГГГГ** в поле **user\_birthday**. Отсортируйте результаты в хронологическом порядке по полю **birthday**. Таблица users

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **first\_name** | **last\_name** | **birthday** |
| 1 | Вероника | Петрова | 1986-04-30 |
| 2 | Светлана | Иванова | 1990-01-22 |
| 3 | Елена | Абрамова | 1994-05-01 |
| 4 | Василиса | Кац | 1991-12-12 |
| 5 | Алёна | Сорокина | 1994-01-07 |
| 6 | Алёна | Федькова | 1989-02-17 |
| 7 | Лиана | Сорокин | 1997-04-29 |
| 8 | Карина | Белая | 1994-06-12 |
| 9 | Анастасия | Дейчман | 1994-06-11 |
| 10 | Юлия | Фёдорова | 1992-02-18 |

**select** first\_name, last\_name, **date\_format**(birthday, "%d.%m.%Y") **as** user\_birthday **from** users **where** **year**(birthday)=1994 **order** **by** birthday

****

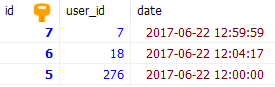
**Задача 7.3.3 Визиты в обед**

Получите из таблицы **visits** все записи с 12:00:00 до 12:59:59 (включительно) за 22 июня 2017 года. Отсортируйте данные в обратном хронологическом порядке (свежие даты сверху).

Таблица visits

|  |  |  |
| --- | --- | --- |
| **id** | **user\_id** | **date** |
| 1 | 1 | 2017-06-21 11:35:17 |
| 2 | 5 | 2017-06-21 12:12:02 |
| 3 | 145 | 2017-06-21 15:43:18 |
| 4 | 147 | 2017-06-22 10:01:00 |
| 5 | 276 | 2017-06-22 12:00:00 |
| 6 | 18 | 2017-06-22 12:04:17 |
| 7 | 7 | 2017-06-22 12:59:59 |
| 8 | 1873 | 2017-06-22 13:00:00 |
| 9 | 18 | 2017-06-22 13:00:01 |
| 10 | 87 | 2017-06-22 14:45:54 |
| 11 | 147 | 2017-06-24 11:35:01 |
| 12 | 971 | 2017-06-24 12:50:44 |
| 13 | 91 | 2017-06-24 11:35:17 |
| 14 | 1355 | 2017-07-22 11:44:41 |
| 15 | 987 | 2017-07-22 12:15:07 |
| 16 | 34 | 2017-07-22 14:00:29 |

**select** \* **from** visits **where** **date**(**date**) = '2017-06-22' **and** **time**(**date**) **BETWEEN** '12:00:00' **and** '12:59:59' **order** **by** **date** **desc**



**Задача 7.3.4 Перенос записей**

Данные о записях пациентов к врачу хранятся в таблице **calendar**.  
Перенесите на 1,5 часа вперед все записи после 13 часов дня (включительно) за 14 мая 2017 года. Таблица calendar

|  |  |  |
| --- | --- | --- |
| **id** | **user\_id** | **visit\_date** |
| 1 | 1 | 2017-05-13 11:00:00 |
| 2 | 5 | 2017-05-13 12:00:00 |
| 3 | 145 | 2017-05-13 15:30:00 |
| 4 | 147 | 2017-05-14 10:00:00 |
| 5 | 276 | 2017-05-14 12:00:00 |
| 6 | 18 | 2017-05-14 13:00:00 |
| 7 | 7 | 2017-05-14 14:00:00 |
| 8 | 1873 | 2017-05-14 15:30:00 |
| 9 | 18 | 2017-05-14 16:30:00 |
| 10 | 147 | 2017-05-16 11:00:00 |
| 11 | 971 | 2017-05-16 12:30:00 |
| 12 | 91 | 2017-05-16 15:30:00 |
| 13 | 1355 | 2017-06-14 11:00:00 |
| 14 | 987 | 2017-06-14 12:30:00 |
| 15 | 34 | 2017-06-14 14:00:00 |

**UPDATE** calendar

**SET** visit\_date = **DATE\_ADD**(visit\_date, **INTERVAL** 1 **HOUR**) **WHERE** visit\_date >= '2017-05-14 13:00:00' **AND** visit\_date <= '2017-05-14 23:59:59';

**UPDATE** calendar

**SET** visit\_date = **DATE\_ADD**(visit\_date, **INTERVAL** 30 **MINUTE**) **WHERE** visit\_date >= '2017-05-14 13:00:00' **AND** visit\_date <= '2017-05-14 23:59:59';

**select** \* **from** calendar



**Задача 7.3.5 Часовые пояса**

В таблице **visits** даты хранятся в часовом поясе по Гринвичу (GMT).  
Выведите **id** посетителей и даты их визита в часовом поясе Москвы ("Europe/Moscow"). Отсортируйте вывод в хронологическом порядке.

Дату следует вывести в формате **ДД.ММ.ГГГГ ЧЧ:ММ** в столбце **visit\_date**. Таблица visits

|  |  |  |
| --- | --- | --- |
| **id** | **user\_id** | **date** |
| 1 | 1 | 2017-06-21 11:35:17 |
| 2 | 145 | 2017-06-21 15:43:18 |
| 3 | 5 | 2017-06-21 12:12:02 |
| 4 | 34 | 2017-07-22 14:00:29 |
| 5 | 147 | 2017-06-22 10:01:00 |
| 6 | 276 | 2017-06-22 12:00:00 |
| 7 | 87 | 2017-06-22 23:12:04 |
| 8 | 18 | 2017-06-22 12:04:17 |
| 9 | 1873 | 2017-06-22 13:00:00 |
| 10 | 7 | 2017-06-22 12:59:59 |
| 11 | 987 | 2017-07-22 12:15:07 |
| 12 | 18 | 2017-06-22 13:00:01 |
| 13 | 91 | 2017-06-24 11:35:17 |
| 14 | 147 | 2017-06-24 11:35:01 |
| 15 | 971 | 2017-06-24 12:50:44 |
| 16 | 1355 | 2017-07-22 11:44:41 |

**select** user\_id, **date\_format** (**date**+**interval** 3 **hour**, "%d.%m.%Y %H:%i") **as** visit\_date **from** visits



**Задача 7.3.6 Гонщики**

В таблице **drivers** хранятся данные о гонщиках спортивного клуба. Выведите фамилию, имя и дни рождения гонщиков, допущенных к соревнованиям, в соответствии с условиями:

1. Допускаются только совершеннолетние мужчины (тем, кому исполнилось 18 лет).

2. Соревнования проходят 8 августа 2018 года.

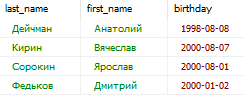
Отсортируйте данные в алфавитном порядке по фамилии и имени.

**Примите** во внимание факт, что по 191 статье гражданского кодекса, совершеннолетие наступает не в день рождения, а на следующий за ним день.

Таблица drivers

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **first\_name** | **last\_name** | **birthday** | **sex** |
| 1 | Дмитрий | Федьков | 2000-01-02 | m |
| 2 | Светлана | Иванова | 1999-03-02 | w |
| 3 | Никита | Трутнев | 2000-08-08 | m |
| 4 | Денис | Кац | 2000-08-09 | m |
| 5 | Алена | Сорокина | 2001-01-02 | w |
| 6 | Василий | Федьков | 2001-01-01 | m |
| 7 | Ярослав | Сорокин | 2000-08-01 | m |
| 8 | Денис | Белый | 2000-12-09 | m |
| 9 | Анатолий | Дейчман | 1998-08-08 | m |
| 10 | Юлия | Фёдорова | 2002-08-08 | w |
| 11 | Вячеслав | Кирин | 2000-08-07 | m |
| 12 | Елизавета | Романова | 2000-08-08 | w |

**select** last\_name, first\_name, birthday **from** drivers **WHERE** sex = 'm' **AND** **DATE\_ADD**(birthday, **INTERVAL** 18 **YEAR**) <= '2018-08-07' **order** **by** last\_name, first\_name

****

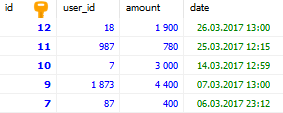
**Задача 7.3.7 Платежи в марте**

Получите из таблицы **payments** последние 5 платежей, которые произошли в марте 2017 года. Выведете номер платежа, id пользователя, сумму и дату (**date**) в формате **ДД.ММ.ГГГГ ЧЧ:ММ**.

Таблица payments

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **user\_id** | **date** | **amount** |
| 1 | 1 | 2017-02-01 11:35:17 | 1100 |
| 2 | 145 | 2017-02-03 15:43:18 | 1200 |
| 3 | 5 | 2017-02-14 12:12:02 | 1200 |
| 4 | 34 | 2017-02-27 14:00:29 | 1300 |
| 5 | 147 | 2017-03-01 10:01:00 | 700 |
| 6 | 276 | 2017-03-04 12:00:00 | 850 |
| 7 | 87 | 2017-03-06 23:12:04 | 400 |
| 8 | 18 | 2017-03-06 12:04:17 | 1400 |
| 9 | 1873 | 2017-03-07 13:00:00 | 4400 |
| 10 | 7 | 2017-03-14 12:59:59 | 3000 |
| 11 | 987 | 2017-03-25 12:15:07 | 780 |
| 12 | 18 | 2017-03-26 13:00:01 | 1900 |
| 13 | 91 | 2017-04-24 11:35:17 | 250 |
| 14 | 147 | 2017-04-24 11:35:01 | 1100 |
| 15 | 971 | 2017-04-27 12:50:44 | 5000 |
| 16 | 1355 | 2017-04-28 11:44:41 | 790 |

**select** id, user\_id, amount, **date\_format**(**date**, "%d.%m.%Y %H:%i") **as** **date** **from** payments **where** **date**(**date**) **between** '2017-03-01' **and** '2017-03-31' **ORDER** **BY** **date** **DESC** **LIMIT** 5



**Задача 7.3.8 Дни рождения в выходные**

Получите из таблицы **users** всех пользователей, которые родились в выходные дни. Отсортируйте данные по полю **birthday** в обратном хронологическом порядке.

Таблица users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **first\_name** | **last\_name** | **birthday** | **sex** |
| 1 | Дмитрий | Федьков | 2000-01-02 | m |
| 2 | Светлана | Иванова | 1999-03-02 | w |
| 3 | Никита | Трутнев | 2000-08-08 | m |
| 4 | Денис | Кац | 2000-08-09 | m |
| 5 | Алена | Сорокина | 2001-01-02 | w |
| 6 | Василий | Федьков | 2001-01-01 | m |
| 7 | Ярослав | Сорокин | 2000-08-01 | m |
| 8 | Денис | Белый | 2000-12-09 | m |
| 9 | Анатолий | Дейчман | 1998-08-08 | m |
| 10 | Юлия | Фёдорова | 2002-08-08 | w |

**select** \* **from** users **where** **dayname**(birthday) **in** ('saturday','sunday') **ORDER** **BY** birthday **DESC**

