\*задания выполнены под синтаксис PostgreSQL

Задание 1.

Описание данных: таблица **ACTIVE\_CLIENTS** содержит ежемесячный срез клиентов банка, которые совершали какие-либо транзакции в данном месяце. Атрибуты: отчетный месяц (report\_month) и идентификатор клиента (client\_id). Считаем, что клиент «оттек» из банка в месяце N, если в месяце N он активен (присутствует в таблице ACTIVE\_CLIENTS) и не активен в месяцы N+1, N+2, N+3.

|  |  |
| --- | --- |
| **REPORT\_MONTH** | **CLIENT\_ID** |
| 2018-01-01 | 1847982357 |
| 2018-01-01 | 938475 |
| 2018-02-01 | 1847982357 |
| 2018-02-01 | 6789998 |
| 2018-03-01 | 67900001 |
| … | … |

**Задание**: вывести количество активных клиентов на каждый месяц; долю клиентов, которые «оттекли» в каждом месяце.

Ответ:

WITH ClientActivity AS (

    SELECT

        report\_month,

        client\_id,

        LEAD(report\_month, 1) OVER (PARTITION BY client\_id ORDER BY report\_month) as month\_1,

        LEAD(report\_month, 2) OVER (PARTITION BY client\_id ORDER BY report\_month) as month\_2,

        LEAD(report\_month, 3) OVER (PARTITION BY client\_id ORDER BY report\_month) as month\_3

    FROM ACTIVE\_CLIENTS),

Mapping AS (

    SELECT

        report\_month,

        client\_id,

        CASE

            WHEN month\_1 IS NULL OR report\_month + INTERVAL '1 month' != month\_1 THEN 1

          WHEN month\_2 IS NULL OR report\_month + INTERVAL '2 months' != month\_2 THEN 1

            WHEN month\_3 IS NULL OR report\_month + INTERVAL '3 months' != month\_3 THEN 1

            ELSE 0

        END AS mapped

    FROM ClientActivity)

SELECT

    ac.report\_month,

    COUNT(DISTINCT ac.client\_id) as total\_active\_clients

    (SUM(mp. mapped) \* 1.0 / COUNT(DISTINCT ac.client\_id)) AS churn\_rate

FROM ACTIVE\_CLIENTS ac

JOIN Mapping mp ON ac.report\_month = mp.report\_month AND ac.client\_id = mp.client\_id

GROUP BY ac.report\_month

ORDER BY ac.report\_month

Задание 2.

Описание данных: таблица **OFFERS** содержит информацию о кредитных предложениях клиентов банка. У каждого предложения есть идентификатор (offer\_id), дата начала действия (offer\_start\_date), дата окончания действия (offer\_expiration\_date).   
**Задание**: вывести количество действующих предложений на каждый день 2018 года.

|  |  |  |
| --- | --- | --- |
| **OFFER\_ID** | **OFFER\_START\_DATE** | **OFFER\_EXPIRATION\_DATE** |
| 83942 | 2017-12-01 | 2018-02-01 |
| 94859 | 2018-05-03 | 2018-10-19 |
| … | … | … |

Ответ:

WITH Dates AS (

    SELECT generate\_series('2018-01-01'::date, '2018-12-31'::date, '1 day') AS dt)

SELECT

    d.dt AS offer\_date,

    COUNT(o.offer\_id) AS active\_offers

FROM Dates d

LEFT JOIN OFFERS o ON d.dt BETWEEN o.offer\_start\_date AND o.offer\_expiration\_date

GROUP BY d.dt

ORDER BY d.dt

Задание 3.

Описание данных: Таблица **CARDS** содержит информацию о всех картах клиентов банка. Атрибуты: идентификатор клиента (client\_id), идентификатор карты (card\_id), дата выдачи карты (open\_date), дата закрытия карты (close\_date), тип карты – дебетовая (DC) или кредитная (СС) (card\_type).   
**Задание**: выбрать из всех работающих на дату 1.09.2018 дебетовых карт клиента ту, которая была выдана последней. Формат выдачи – client\_id, card\_id.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CLIENT\_ID** | **CARD\_ID** | **OPEN\_DATE** | **CLOSE\_DATE** | **CARD\_TYPE** |
| 1232110 | 49582985729 | 2019-01-12 | NULL | DC |
| 234235 | 48574092749 | 2017-03-29 | 2018-09-01 | CC |
| … | … | … | … | … |

Ответ:

SELECT

  client\_id,

  card\_id

FROM CARDS

WHERE card\_type = 'DC'

  AND open\_date <= '2018-09-01'

  AND (close\_date IS NULL OR close\_date > '2018-09-01')

ORDER BY open\_date DESC

LIMIT 1

Задание 4.

Описание данных: Таблица **BONUS** содержит информацию о начислениях бонусных баллов за покупки по программе лояльности. Атрибуты: идентификатор клиента (client\_id), дата начисления бонусов (bonus\_date), количество начисленных бонусов (bonus\_cnt), mcc код транзакции, за которую начислены бонусы (mcc\_code). Таблица **MCC\_CATEGORIES** является справочником mcc-кодов. Атрибуты: mcc-код (mcc\_code), категория (например, супермаркеты, транспорт, аптеки и пр., mcc\_category)   
**Задание**: Вывести 1000 клиентов, которые первыми набрали 1000 бонусных баллов за покупки в категории «Авиабилеты» и «Отели».

|  |  |  |  |
| --- | --- | --- | --- |
| **CLIENT\_ID** | **BONUS\_DATE** | **BONUS\_CNT** | **MCC\_CODE** |
| 1232110 | 2018-01-01 | 12 | 3617 |
| 234235 | 2018-06-17 | 5 | 5931 |
| … | … | … | … |

|  |  |
| --- | --- |
| **MCC\_CODE** | **MCC\_CATEGORY** |
| 3031 | Авиабилеты |
| 5735 | Музыка |
| … | … |

Ответ:

WITH ClientBonus AS (

    SELECT

        b.client\_id,

        b.bonus\_date,

        SUM(b.bonus\_cnt) OVER (PARTITION BY b.client\_id ORDER BY b.bonus\_date) AS sum\_bonus

    FROM BONUS b

    JOIN MCC\_CATEGORIES mcc ON b.mcc\_code = mcc.mcc\_code

    WHERE mcc.mcc\_category IN ('Авиабилеты', 'Отели'))

SELECT

    client\_id

FROM

(SELECT

        client\_id,

        MIN(bonus\_date) AS achieved\_date

    FROM ClientBonus

    WHERE sum\_bonus >= 1000

    GROUP BY client\_id) df

ORDER BY achieved\_date

LIMIT 1000

Задание 5.

Есть таблица курсов валют ЦБ, состоящая из трех полей:

CODE [ISO код валюты],

RATE [Значение курса],

VALUE\_DAY [Дата начала действия курса].

Данные в нее записываются каждый раз при изменении курса какой-либо валюты. (Обратите внимание не каждый день, а в день, когда ЦБ поменял курс)

Задача: получить курс валюты «USD» на заданную (любую) дату.

Ответ:

SELECT

    RATE

FROM Exchange\_Rates

WHERE

    CODE = 'USD' AND VALUE\_DAY <= '2024-09-27'

ORDER BY

    VALUE\_DAY DESC

LIMIT 1