**Задача 1: Даны два реестра, содержащие идентификатор клиента и идентификатор ценной бумаги. Реестры были заполнены вручную пользователями. Необходимо написать запрос, который позволит понять, идентичны ли реестры. Порядок хранения данных в реестрах значения не имеет.**

CREATE TABLE REG\_2021\_11 (PERSON\_ID ЧИСЛО, ASSET\_ID ЧИСЛО);

CREATE TABLE REG\_2021\_12 (PERSON\_ID ЧИСЛО, ASSET\_ID ЧИСЛО);

Пример данных:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| REG\_2021\_11:   |  |  | | --- | --- | | **PERSON\_ID** | **ASSET\_ID** | | 1 | 1 | | 2 | 2 | | 2 | 2 | | 3 | 3 | | 4 | 4 | |  |
| REG\_2021\_12:   |  |  | | --- | --- | | **PERSON\_ID** | **ASSET\_ID** | | 1 | 1 | | 2 | 2 | | 3 | 3 | | 3 | 3 | | 4 | 4 | |  |

**Задача 2: Имеется таблица с транзакциями клиента. Необходимо написать запрос, который вернет все транзакции за 15 и 16 неделю каждого года и общую сумму транзакций за эти же недели за все годы по этому клиенту. Суммы транзакции и общая сумма должны быть выведены в следующем формате:**

1. **Разделить целой и дробной части ‘,’ (запятая)**
2. **Разделить тысяч ‘ ’(пробел)**
3. **2 знака в дробной части**
4. **Не менее 1 знака в целой части.**

**При определении номера недели необходимо ориентироваться на правила нумерации номеров недель в РФ.**

CREATE TABLE T (CL\_ID ЧИСЛО, NM СТРОКА, DT ДАТА, SUMM ЧИСЛО);

Пример данных:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | **CL\_ID** | **NM** | **DT** | **SUMM** | | 1 | Иванов | 01.01.2022 | 10.12 | | 1 | Иванов | 12.02.2022 | 120.21 | | 1 | Иванов | 16.04.2021 | 0.22 | | 1 | Иванов | 15.04.2022 | 356.1 | | 1 | Иванов | 21.04.2021 | 12.1 | | 2 | Петров | 18.04.2020 | 100 | | 2 | Петров | 13.04.2019 | 25512.12 | |
|  |

Пример результата для одной транзакции:

Вместо «ХХ» необходимо отобразить номер соответствующей недели.

Вместо «YYYY» необходимо отобразить год, в котором была произведена транзакция, к примеру 2021

|  |  |  |  |
| --- | --- | --- | --- |
| **Клиент** | **Неделя** | **Сумма транзакции** | **Общая сумма за неделю за все годы** |
| Иванов | ХХ неделя YYYY | 0,00 | 0,00 |

**Задача 3: Есть таблица, содержащая организационную структуру компании. Необходимо написать запрос, который выведет структуру компании в виде дерева, исключив из нее отдел рекламы и подчиненные ему подразделения. Для каждого подразделения необходимо вывести название родительского подразделения. Данные должны быть отсортированы по наименованию в алфавитном порядке с учетом иерархии.**

CREATE TABLE T (ID ЧИСЛО, PID ЧИСЛО, NAM СТРОКА);

Пример данных:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **ID** | **PID** | **NAM** | | 1 |  | Банк | | 2 | 1 | Департамент IT | | 3 | 1 | Департамент безопасности | | 4 | 2 | Управление развития | | 5 | 4 | Отдел рекламы | | 6 | 5 | Группа баннеров | | 7 | 4 | Отдел развития | |  |

Требуемый результат:

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **PID** | **NAM** | **PARENT\_NAM** |
| 1 |  | Банк |  |
| 2 | 1 | Департамент IT | Банк |
| 4 | 2 | Управление развития | Департамент IT |
| 7 | 4 | Отдел развития | Управление развития |
| 3 | 1 | Департамент безопасности | Банк |

**Задача 4: Имеется таблица с клиентами и таблица с продуктами клиентов. Необходимо написать запрос, который по определенному клиенту сформирует XML со списком продуктов клиента (в значение тегов продуктов необходимо вывести поле status).**

Пример данных:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Таблица Clients:   |  |  | | --- | --- | | **CL\_ID** | **NAME** | | 1 | Иванов | | 2 | Петров | | 3 | Сидоров | |  |
| Таблица Products:   |  |  |  |  | | --- | --- | --- | --- | | **CL\_ID** | **Prod\_id** | **Status** | **Name** | | 1 | 1 | 0 | Кредит | | 1 | 2 | 1 | Депозит | | 1 | 3 | 1 | Акция | | 1 | 4 | 0 | Облигация | | 2 | 1 | 1 | Кредит | | 2 | 2 | 0 | Депозит | |  |

Требуемый результат:

<?xml version = '1.0'>

<Client name = "Иванов">

<Product name ="Кредит">0</Product>

<Product name ="Депозит">1</Product>

<Product name ="Акция">1</Product>

<Product name ="Облигация">0</Product>

</Client>

**Задача 5: Имеется таблица c данными по платежным документам. Необходимо написать запрос, который выведет все документы того типа, которого за все время было по сумме больше всего. Если таких типов несколько, то вывести все такие типы. Для каждой строки результата вывести промежуточную сумму платежей данного типа от самого раннего до текущего платежа включительно.**

CREATE TABLE Payments  
(ID ЧИСЛО, PAY\_TYPE ЧИСЛО, PAY\_TYPE\_LABEL СТРОКА, PAY\_DATE ДАТА, PAY\_SUM ЧИСЛО)

Пример данных:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **PAY\_TYPE** | **PAY\_TYPE\_LABEL** | **PAY\_DATE** | **PAY\_SUM** |
| 1 | 2 | Перевод денежных средств | 01.01.2022 | 600 |
| 2 | 3 | Оплата услуг | 10.01.2022 | 1100 |
| 3 | 1 | Комиссия | 01.01.2022 | 100 |
| 4 | 1 | Комиссия | 02.01.2022 | 200 |
| 5 | 3 | Оплата услуг | 01.03.2022 | 1200 |
| 6 | 2 | Перевод денежных средств | 01.02.2022 | 700 |
| 7 | 1 | Комиссия | 03.01.2022 | 300 |
| 8 | 2 | Перевод денежных средств | 01.04.2022 | 800 |
| 9 | 3 | Оплата услуг | 01.05.2022 | 1300 |
| 10 | 1 | Комиссия | 01.02.2022 | 400 |
| 11 | 2 | Перевод денежных средств | 01.05.2022 | 900 |
| 12 | 3 | Оплата услуг | 05.05.2022 | 1400 |
| 13 | 3 | Оплата услуг | 01.06.2022 | 1500 |
| 14 | 2 | Перевод денежных средств | 01.06.2022 | 1000 |
| 15 | 1 | Комиссия | 01.02.2022 | 500 |

Требуемый результат:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **PAY\_TYPE** | **PAY\_TYPE\_LABEL** | **PAY\_DATE** | **PAY\_SUM** | **SM** |
| 2 | 3 | Оплата услуг | 10.01.2022 | 1100 | 1100 |
| 5 | 3 | Оплата услуг | 01.03.2022 | 1200 | 2300 |
| 9 | 3 | Оплата услуг | 01.05.2022 | 1300 | 3600 |
| 12 | 3 | Оплата услуг | 05.05.2022 | 1400 | 5000 |
| 13 | 3 | Оплата услуг | 01.06.2022 | 1500 | 6500 |

**Задача 6: Имеется таблица, содержащая информацию по клиентам. Необходимо написать запрос, который подготовит список клиентов для рассылки поздравительных писем, у которых в ближайшие N дней от заданной даты будет день рождения.**

CREATE TABLE T\_CL (ID ЧИСЛО, NAME СТРОКА, BIRTH\_DAY ДАТА)

Пример данных:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **T\_CL**:   |  |  |  | | --- | --- | --- | | **ID** | **NAME** | **BIRTH\_DAY** | | 1 | Клиент 1 | 05.01.1965 | | 2 | Клиент 2 | 25.01.1976 | | 3 | Клиент 3 | 07.02.1978 | | 4 | Клиент 4 | 17.03.1979 | | 5 | Клиент 5 | 26.08.1979 | | 6 | Клиент 6 | 05.12.1962 | |  |

Требуемый результат при заданной дате 01.12.2021 и количестве дней 50:

|  |  |  |
| --- | --- | --- |
| **ID** | **NAME** | **BIRTH\_DAY** |
| 1 | Клиент 1 | 05.01.1965 |
| 6 | Клиент 6 | 05.12.1962 |

**Задача 7: Дана таблица с этапами заявок**

CREATE TABLE TBL\_ORDER\_ROW

( STAGE\_ID ЧИСЛО -- Идентификатор этапа

, ORDER\_ID ЧИСЛО -- Идентификатор заявки

, BEGIN\_DATE ДАТА -- Дата начала выполнения этапа

, END\_DATE ДАТА -- Дата завершения выполнения этапа

, STAGE\_TYPE\_ID ЧИСЛО -- Тип этапа

)

**Необходимо написать программу, которая за один проход по таблице вычисляет:**

**MIN – минимальное время обработки заявки на этапе;**

**MAX – максимальное время обработки заявки на этапе;**

**AVG – среднее значение времени обработки заявки на этапе, из которого исключены минимальные и максимальные значения времени обработки (если таковое значение отсутствует, то вывести “NO”).**

**Полученные результаты необходимо сохранить в таблицу логов:**

CREATE TABLE TMP\_CLOB

( TIME\_LOG ДАТА с часовым поясом (timestamp) -- Дата начала обработки

, STAGE\_TYPE\_ID ЧИСЛО -- Тип этапа

, TEXT СТРОКА -- Примечание

)

**Уникальный ключ по столбцам** TIME\_LOG, STAGE\_TYPE\_ID

Пример данных в таблице TBL\_ORDER\_ROW:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STAGE\_ID** | **ORDER\_ID** | **BEGIN\_DATETIME** | **END\_DATETIME** | **STAGE\_TYPE\_ID** |
| 437855 | 77832 | 25.11.2021 00:12:19 | 25.11.2021 00:12:26 | 5571 |
| 437856 | 77832 | 25.11.2021 00:12:26 | 25.11.2021 00:12:27 | 5572 |
| 437857 | 77832 | 25.11.2021 00:12:33 | 25.11.2021 00:12:34 | 5573 |
| 437864 | 77833 | 25.11.2021 00:14:49 | 25.11.2021 00:14:56 | 5571 |
| 437865 | 77833 | 25.11.2021 00:14:56 | 25.11.2021 00:14:56 | 5572 |
| 437866 | 77833 | 25.11.2021 00:15:04 | 25.11.2021 00:15:04 | 5573 |
| 437873 | 77834 | 25.11.2021 00:24:33 | 25.11.2021 00:24:41 | 5571 |
| 437874 | 77834 | 25.11.2021 00:24:41 | 25.11.2021 00:24:41 | 5572 |
| 437875 | 77834 | 25.11.2021 00:24:49 | 25.11.2021 00:24:49 | 5573 |
| 437882 | 77835 | 25.11.2021 00:25:22 | 25.11.2021 00:25:22 | 5571 |
| 437883 | 77835 | 25.11.2021 00:25:22 | 25.11.2021 00:25:22 | 5572 |
| 437884 | 77835 | 25.11.2021 00:25:22 | 25.11.2021 00:25:23 | 5573 |
| 437891 | 77836 | 25.11.2021 00:44:04 | 25.11.2021 00:44:11 | 5571 |
| 437892 | 77836 | 25.11.2021 00:44:11 | 25.11.2021 00:44:11 | 5572 |
| 437893 | 77836 | 25.11.2021 00:44:18 | 25.11.2021 00:44:19 | 5573 |
| 437900 | 77837 | 25.11.2021 00:48:03 | 25.11.2021 00:48:11 | 5571 |
| 437901 | 77837 | 25.11.2021 00:48:11 | 25.11.2021 00:48:11 | 5572 |
| 437902 | 77837 | 25.11.2021 00:48:18 | 25.11.2021 00:48:19 | 5573 |
| 437911 | 77840 | 25.11.2021 01:29:19 | 25.11.2021 01:29:26 | 5571 |
| 437912 | 77840 | 25.11.2021 01:29:26 | 25.11.2021 01:29:26 | 5572 |
| 437913 | 77840 | 25.11.2021 01:29:34 | 25.11.2021 01:29:34 | 5573 |
| 437920 | 77841 | 25.11.2021 01:36:48 | 25.11.2021 01:36:55 | 5571 |
| 437921 | 77841 | 25.11.2021 01:36:55 | 25.11.2021 01:36:56 | 5572 |
| 437922 | 77841 | 25.11.2021 01:37:04 | 25.11.2021 01:37:04 | 5573 |

Пример данных в логе:

|  |  |  |
| --- | --- | --- |
| **TIME\_LOG** | **STAGE\_TYPE\_ID** | **TEXT** |
| 25.11.21 20:34:40,355577 +03:00 | 5571 | MIN=0; MAX=8; AVG=7 |
| 25.11.21 20:34:40,355577 +03:00 | 5572 | MIN=0; MAX=1; AVG= NO |
| 25.11.21 20:34:40,355577 +03:00 | 5573 | MIN=0; MAX=1; AVG= NO |