1. **В***ывести 1000 случайных чисел от 1 до 1000, таких что не повторяются в этой последовательности, больше чем 3 раза.*

**2. Имеется таблица без первичного ключа. Известно, что в таблице имеется задвоение данных. Необходимо удалить дубликаты из таблицы.**

create table **t** (**a** number, **b** number);

Пример данных:

**a b**

1 1

2 2

2 2

3 3

3 3

3 3

Требуемый результат:

**a b**

1 1

2 2

3 3

**3. XML**

Имеется

xmltype('

<root>

  <row>

    <col>v11</col>

    <col>v12</col>

    <col>v13</col>

    <col>v14</col>

  </row>

  <row>

    <col>v21</col>

    <col>v22</col>

    <col>v23</col>

    <col>v24</col>

  </row>

</root>')

Необходимо:

а) Получить выборку

C1   C2   C3   C4

---- ---- ---- ----

v11  v12  v13  v14

v21  v22  v23  v24

Условия: количество узлов row может варьироваться, col всегда статично = 4 шт в пределах row

b) Получить в виде результата колонку с типом xmltype SQL запроса со следующей структурой:

<root>

   <data row="1" col="1">v11</data>

   <data row="1" col="2">v12</data>

   <data row="1" col="3">v13</data>

   <data row="1" col="4">v14</data>

   <data row="2" col="1">v21</data>

   <data row="2" col="2">v22</data>

   <data row="2" col="3">v23</data>

   <data row="2" col="4">v24</data>

</root>

Реализовать данный запрос не используя XSLT трансформацию.

Условия: количество узлов row и col может варьироваться (прим. это более сложный пример, можно вернуть условие что количество col всегда статично = 4 шт в пределах row)

c) Получить в виде результата колонку с типом xmltype SQL запроса со следующей структурой:

<root>

   <data row="1" col="1">v11</data>

   <data row="1" col="2">v12</data>

   <data row="1" col="3">v13</data>

   <data row="1" col="4">v14</data>

   <data row="2" col="1">v21</data>

   <data row="2" col="2">v22</data>

   <data row="2" col="3">v23</data>

   <data row="2" col="4">v24</data>

</root>

Реализовать данный запрос используя XSLT трансформацию.

Условия: количество узлов row и col может варьироваться

**4. Коллекции**

Имеется декларация типа:

CREATE OR REPLACE TYPE TNUM as table of number;

Необходимо написать реализацию функции, возвращающая в качестве результата заполненный массив имеющий тип TNUM с значениями от 1..1000

**5. Регулярные выражения**

Задана строка '1,2,3,4', необходимо, используя регулярные выражения в запросе получить результат:

C1 C2 C3 C4

-- -- -- --

1  2  3  4

**6. Использование pipelined функции**

Имеется таблица dept со следующей структурой:

Name   Type         Nullable Default Comments

------ ------------ -------- ------- --------

DEPTNO NUMBER

DNAME  VARCHAR2(14) Y

LOC    VARCHAR2(13) Y

Необходимо реализовать функцию PL/SQL которая будет возвращать выборку из таблицы dept заданную минимальным и максимальным значением поля DEPTNO. Реализуемая функция должна использовать метод pipelined.

**7. Отчёт для отдела маркетинга**

Имеется база данных по клиентам, имеющая структуру:

|  |  |  |
| --- | --- | --- |
| Клиенты | | |
| ID | number |  |
| NAME | Varchar2 | Наименование |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Контакты | | |
| ID | number |  |
| CLIENT\_ID | number | FK CLIENT |
| C\_TYPE | number | Тип контакта 1-телефон 2-email |
| C\_INFO | varchar2 | Контакт – телефон либо адрес email |
| CREATED | date | Дата внесения в базу |
| ACTIVE | Char(1) | Y/N активный или архив |

|  |  |  |
| --- | --- | --- |
| Адреса | | |
| ID | number |  |
| CLIENT\_ID | number | FK CLIENT |
| A\_TYPE | number | Тип адреса 1-домашний 2-регистрации 3- фактический |
| CITY | varchar2 | Город |
| STREET | varchar2 | Улица |
| HOUSE | varchar2 | Дом |
| FLAT | varchar2 | Квартира |
| CREATED | date | Дата внесения в базу |
| ACTIVE | Char(1) | Y/N активный или архив |

Отделу маркетинга требуется сводная выгрузка по клиентам, с гранулярностью до клиента, при этом для каждого клиента в выборке должны быть «лучшие» адрес, телефон и адрес электронной почты. То есть, в результирующей выборке по каждому клиенту есть только одна строка. При этом:

1. Лучший адрес отбирается по приоритету фактический > регистрации > домашний, при наличии нескольких адресов одного приоритета выбирается наиболее полный (заполнено больше из перечня атрибутов city-street-house-flat, при равенстве по заполненности выбирается последний по дате внесения в базу.
2. Лучший телефон это последний по дате внесения в базу
3. Лучший email это первый по дате внесения в базу
4. Данные по контактам и адресам – не архивные

8. **Аналитические функции**

Создать таблицу из произвольного количества >1000 строк с четырьмя полями: номер строки и четыре разных **целых** псевдослучайных числа в интервале от 1 до 100.

Вывести запросом:

- для первой строки – номер, первое число, второе число из следующей строки, третье число из следующей строки

- для последней строки – номер, первое число, второе число из предыдущей строки, третье число из предыдущей строки

- если первое число четное – номер, первое число, второе число из предыдущей строки, третье число из следующей строки

- если первое число нечетное – номер, первое число, второе число из следующей строки, третье число из предыдущей строки

9. **Аналитические функции**

Создать таблицу из произвольного количества >1000 строк с двумя псевдослучайными моментами времени из одних суток: начало телефонного разговора, конец телефонного разговора. Длительность разговора не должна превышать 200 секунд. Момент начала и конца разговора включаются в разговор.

Написать функцию, которая для произвольного момента времени определяет число разговоров в данный момент времени.

Написать запрос, который определит максимальное число одновременных разговоров в течение суток