**Задание 1.**

(

Библиотека **ORION**

libname ORION **disk ‘/var/data/mipt’;**

)

**a.** Напишите шаг DATA, считывающий **orion.customer** и создающий **work.birthday**.

**b.** В шаге DATA создайте три новые переменные: **Bday2012**, **BdayDOW2012** и **Age2012**.

 **Bday2012** вычисляется при помощи функции MDY, используя месяц и число даты рождения клиента **Birth\_Date**, и константу **2012**, задающую год.

 **BdayDOW2012** - день недели, посчитанный для даты **Bday2012**.

 **Age2012** - возраст клиента в 2012 году. Отнимите дату **Birth\_Date** от **Bday2012** и разделите результат на 365.25.

**c.** Включите в новый набор данных только указанные переменные: **Customer\_Name**, **Birth\_Date**, **Bday2012**, **BdayDOW2012** и **Age2012**.

**d.** Отформатируйте **Bday2012**, чтобы она выводилась в виде 01Jan2012. **Age2012** нужно отформатировать так, чтобы она выводилась без десятичных знаков.

**e.** Напишите шаг PROC PRINT для создания отчёта. Результаты должны содержать 77 наблюдений.

**Задание 2.**

Напишите шаг DATA, считывающий набор данных **orion.customer\_dim** и создающий **work.season**.

**b.** Создайте в нём две новые переменные: **Promo** и **Promo2**.

Значение переменной **Promo** основано на квартале, в котором клиент празднует свой день рождения.

 Если клиент родился в первом квартале, тогда **Promo** имеет значение *Winter*.

 Если клиент родился во втором квартале, тогда **Promo** имеет значение *Spring*.

 Если клиент родился в третьем квартале, тогда **Promo** имеет значение *Summer*.

 Если клиент родился в четвёртом квартале, тогда **Promo** имеет значение *Fall*.

Значение **Promo2** зависит от возраста клиента.

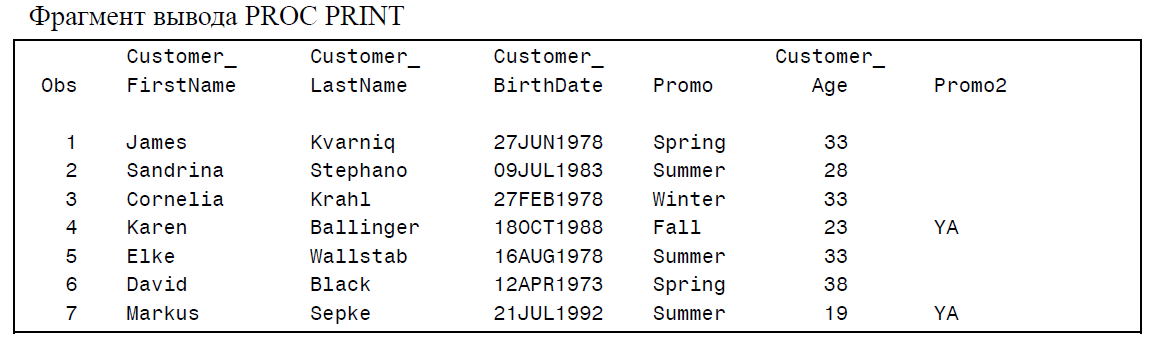
• Для молодых клиентов, чей возраст находится между 18 и 25 годами, значение Promo2 равно YA.

• Для пожилых, чей возраст превышает 65 лет, значение Promo2 равно Senior.

**Promo2** должна иметь пропущенное значение для всех остальных клиентов.

**c.** Новый набор данных должен содержать только переменные **Customer\_FirstName**, **Customer\_LastName**, **Customer\_BirthDate**, **Customer\_Age**, **Promo** и **Promo2**.

**d.** Создайте отчёт, показанный ниже. Результаты должны содержать 77 наблюдений.



**Задание 3.**

Напишите шаг DATA, считывающий набор данных **orion.orders** и создающий **work.ordertype**.

**f.** Создайте новую переменную, **DayOfWeek**, которая вычисляется как день недели от даты **Order\_Date**.

**g.** Создайте новую переменную **Type**, принимающую следующие значения:

 *Retail Sale*, если **Order\_Type** равен *1*

 *Catalog Sale*, если **Order\_Type** равен *2*

 *Internet Sale*, если **Order\_Type** равен *3*.

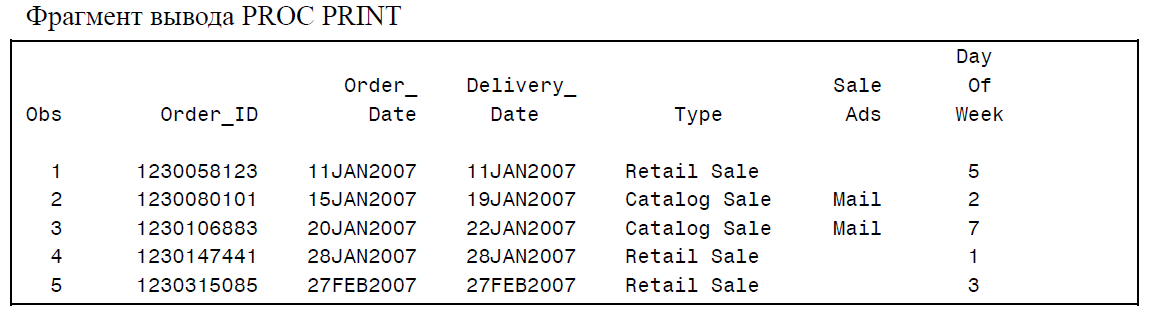
**h.** Создайте новую переменную **SaleAds**, которая принимает указанные значения:

 *Mail*, если **Order\_Type** равен *2*

 *Email*, если **Order\_Type** равен *3*.

**i.** Не включайте в новый набор данных переменные **Order\_Type**, **Employee\_ID** и **Customer\_ID.**

**j.** Создайте отчёт, показанный ниже. Результаты должны содержать 490 наблюдений.



**Задание 4.** Создание основной таблицы.

Создать таблицу WORK.CR\_TABLE на шаге PROC SQL со следующей структурой:

|  |  |  |
| --- | --- | --- |
| VARIABLE\_NM | VARIABLE\_TYPE | VARIABLE\_FORMAT |
| CUSTOMER\_ID | NUMERIC | NONE |
| CUSTOMER\_NAME | CHAR(32) | NONE |
| CUSTOMER\_JOB | CHAR(64) | NONE |
| REGIST\_DATE | NUMERIC | DATE9. |

**Задание 5.** Создать таблицу WORK.SOURCE шагом DATA с такой же структурой.  
Сгенерировать следующие строки:  
1, Bob, Student, 25jan1995

2, James, Director, 17JUN1983.

**Задание 6.** Наполнение основной таблицы.

Добавить в таблицу WORK.CR\_TABLE данные из таблицы WORK.SOURCE на шаге PROC SQL.

После чего необходимо дополнить таблицу еще одной строкой (PROC SQL)

3, John, Programmer, 01JAN2001.

**Задание 7.** Изменение структуры таблицы.

Добавить ярлык (label) для полей CUSTOMER\_ID, CUSTOMER\_JOB – ID сотрудника, Профессия сотрудника за один шаг.

**Задание 8.** Добавление нового поля в таблицу.

5.1. Добавить новое поле на шаге PROC SQL – customer\_age (num).

5.2. Апдейтнуть таблицу, присвоив новому столбцу customer\_age следующее выражение – (дата\_текущая – REGIST\_DATE). Результат должен отображать количество лет\*

**Задание 9.** Удалить выбранные строки.

Удалить строки по следующему условию – customer\_age>25.

**Задание 10.**  Выполнить код, создающий таблицу (с использованием оператора datalines)

data work.suppliers;

infile datalines delimiter=',';

input Supplier\_Category $

Supplier\_Country $

Supplier\_Name $

Salary :8.

Hier\_date :date9.

;

datalines;

Sports,DK,Jack,1200,01OCT2015

Sports,DK,Boris,800,01JAN2015

Sports,DK,James,1900,01DEC2014

Trading,US,Bill,2000,01DEC2016

Outdoor,AU,Peter,2000,01APR2018

Outdoor,AU,Kate,3000,01MAR2018

Sports,DK,Sandy,10000,01APR2018

Sports,DK,Marple,770,01FEB2017

Sports,DK,Jinny,1450,01MAY2019

Outdoor,AU,Privet,1650,01AUG2020

Sports,DK,Mishel,1500,01MAR2018

Sasteria,ES,Annet,250,01OCT2017

Trading,US,Julien,1345,01JUL2010

;

run;

**10.1.** Создать новую таблицу work. category\_aggr, в которую записать среднее значение Salary для каждой категории (Supplier\_Category).

Создать таблицу work.country\_aggr со средним значением Salary для Supplier\_Country.

Создать таблицу work.report\_suppliers на основе соединения таблиц suppliers, category\_aggr, country\_aggr в библиотеке WORK, добавив ко всем столбцам таблицы suppliers полученные агрегаты.  
  
**10.2.** На основе report\_suppliers создать новую таблицу work.report\_supp\_actual, в которую записать значения, у которых максимальное значение Hier\_date в группировке по полю Supplier\_Category (реализация должна быть в один запрос (как вариант, использовать подзапросы).

**Задание 11.** Выполнить код для формирования двух таблиц

data Customers;

length Customer\_ID 8

Name

Address $64;

Customer\_ID=00001;

Name='ООО "Ромашка+"';

Address='None';

output;

Customer\_ID=00004;

Name='ОАО "РЖД"';

Address='None';

output;

Customer\_ID=00002;

Name='ЗАО "ИнтерСофт"';

Address='None';

output;

Customer\_ID=00003;

Name='Umbrella Corp.';

Address='None';

output;

Customer\_ID=00006;

Name='Waterfall Enterprise';

Address='None';

output;

Customer\_ID=00005;

Name='ООО "Ромашка Плюс Плюс"';

Address='None';

output;

run;

data Tickets;

length Ticket\_ID

Customer\_ID

Status

Open\_DT

Close\_DT 8;

format open\_dt close\_dt date9.;

Ticket\_ID=000001;

Customer\_ID=00001;

Status=1;

Open\_DT='01jan2015'd;

Close\_DT=.;

output;

Ticket\_ID=000002;

Customer\_ID=00001;

Status=0;

Open\_DT='01jan2015'd;

Close\_DT='10jan2015'd;

output;

Ticket\_ID=000003;

Customer\_ID=00004;

Status=1;

Open\_DT='01jan2015'd;

Close\_DT=.;

output;

Ticket\_ID=000004;

Customer\_ID=00001;

Status=0;

Open\_DT='02jan2015'd;

Close\_DT='10jan2015'd;

output;

Ticket\_ID=000005;

Customer\_ID=00002;

Status=0;

Open\_DT='03jan2015'd;

Close\_DT='11jan2015'd;

output;

Ticket\_ID=000006;

Customer\_ID=00002;

Status=0;

Open\_DT='04jan2015'd;

Close\_DT='12jan2015'd;

output;

Ticket\_ID=000007;

Customer\_ID=00002;

Status=0;

Open\_DT='05jan2015'd;

Close\_DT=.;

output;

Ticket\_ID=000008;

Customer\_ID=00003;

Status=1;

Open\_DT='06jan2015'd;

Close\_DT=.;

output;

Ticket\_ID=000009;

Customer\_ID=00003;

Status=0;

Open\_DT='07jan2015'd;

Close\_DT='12jan2015'd;

output;

Ticket\_ID=000010;

Customer\_ID=00003;

Status=0;

Open\_DT='08jan2015'd;

Close\_DT='16jan2015'd;

output;

Ticket\_ID=000011;

Customer\_ID=00001;

Status=1;

Open\_DT='09jan2015'd;

Close\_DT=.;

output;

run;

**Задача 11.1**

Вывести список клиентов у которых ВСЕ тикеты закрыты (status = 1)

**Задача 11.2**

Вывести клиентов у которых >2 открытых тикетов