## Задача №1

Дано 2 таблицы:

1) employee - справочник сотрудников

2) tasks - задачи, назначенные на сотрудников

необходимо написать запрос, который покажет всех сотрудников, у кого в работе менее трех задач

Результат предоставить в виде: имя сотрудника, кол-во задач в работе.

drop table if exists #employee

create table #employee

(

id smallint,

emp\_name varchar(255),

rating int

)

insert into #employee values (1,'Vasya',5)

insert into #employee values (2,'Vasya',3)

insert into #employee values (3,'Petya',9)

insert into #employee values (4,'Igor',4)

insert into #employee values (5,'Nikita',2)

drop table if exists #tasks

create table #tasks

(

id smallint,

assignee\_id smallint,

rating smallint

)

insert into #tasks values (1,1,5)

insert into #tasks values (2,1,4)

insert into #tasks values (3,2,2)

insert into #tasks values (4,5,5)

insert into #tasks values (5,5,1)

insert into #tasks values (6,5,7)

insert into #tasks values (7,5,9)

insert into #tasks values (8,5,3)

insert into #tasks values (9,5,3)

insert into #tasks values (10,5,5)

insert into #tasks values (11,2,7)

insert into #tasks values (12,3,8)

insert into #tasks values (13,3,8)

insert into #tasks values (14,2,7)

insert into #tasks values (15,3,9)

insert into #tasks values (16,5,4)

insert into #tasks values (17,1,5)

### Решение:

select emp\_name, count(assignee\_id) as task\_count

from #employee e

full outer join #tasks t

on e.id = t.assignee\_id

group by e.id, emp\_name

### Ответ:

|  |  |
| --- | --- |
| **emp\_name** | **task\_count** |
| Igor | 0 |
| Nikita | 8 |
| Petya | 3 |
| Vasya | 3 |
| Vasya | 3 |

## Задача №2

Дано 2 таблицы:

1) payments\_principal - платежи по основному долгу

2) payments\_interest - платежи по процентам

Необходимо написать запрос, который покажет сумму платежей по основному долгу и процентам.

Суммы необходимо считать только по тем договорам, которые платили и основной долг и проценты.

Результат предоставить в виде: id договора, сумма платежей по основному долгу, сумма платежей по процентам.

drop table if exists #payments\_principal

create table #payments\_principal

(

payment\_dt date,

agr\_id smallint,

payment\_sum int

)

drop table if exists #payments\_interest

create table #payments\_interest

(

payment\_dt date,

agr\_id smallint,

payment\_sum int

)

insert into #payments\_principal values ('2019-05-11',31,8281)

insert into #payments\_principal values ('2019-05-12',7,4622)

insert into #payments\_principal values ('2019-05-13',5,7686)

insert into #payments\_principal values ('2019-07-01',1,9917)

insert into #payments\_principal values ('2019-07-23',1,6534)

insert into #payments\_principal values ('2019-08-20',64,4336)

insert into #payments\_principal values ('2019-08-24',3,7464)

insert into #payments\_principal values ('2019-08-25',9,8505)

insert into #payments\_principal values ('2019-08-27',1,9857)

insert into #payments\_principal values ('2019-07-07',7,6294)

insert into #payments\_principal values ('2019-07-17',7,3182)

insert into #payments\_principal values ('2019-08-28',4,9708)

insert into #payments\_principal values ('2019-08-29',4,8632)

insert into #payments\_principal values ('2019-08-30',3,8303)

insert into #payments\_principal values ('2019-09-01',7,3141)

insert into #payments\_principal values ('2019-08-25',1,9139)

insert into #payments\_principal values ('2019-08-25',2,7624)

insert into #payments\_principal values ('2019-09-01',7,3793)

insert into #payments\_principal values ('2019-09-01',3,3260)

insert into #payments\_principal values ('2019-08-21',5,9002)

insert into #payments\_principal values ('2019-08-22',2,5500)

insert into #payments\_principal values ('2019-05-12',7,4622)

insert into #payments\_principal values ('2019-08-23',2,3980)

insert into #payments\_principal values ('2019-08-29',2,5849)

insert into #payments\_interest values ('2019-05-11',31,98)

insert into #payments\_interest values ('2019-05-12',7,90)

insert into #payments\_interest values ('2019-05-13',5,39)

insert into #payments\_interest values ('2019-07-01',1,82)

insert into #payments\_interest values ('2019-07-23',1,59)

insert into #payments\_interest values ('2019-08-20',50,96)

insert into #payments\_interest values ('2019-08-24',3,1)

insert into #payments\_interest values ('2019-08-25',9,22)

insert into #payments\_interest values ('2019-08-27',1,95)

insert into #payments\_interest values ('2019-07-07',7,79)

insert into #payments\_interest values ('2019-07-17',7,72)

insert into #payments\_interest values ('2019-08-28',4,61)

insert into #payments\_interest values ('2019-08-29',4,49)

insert into #payments\_interest values ('2019-08-30',3,78)

insert into #payments\_interest values ('2019-09-01',7,29)

insert into #payments\_interest values ('2019-08-25',1,88)

insert into #payments\_interest values ('2019-08-25',2,77)

insert into #payments\_interest values ('2019-09-01',6,6)

insert into #payments\_interest values ('2019-09-01',3,18)

insert into #payments\_interest values ('2019-08-21',5,15)

insert into #payments\_interest values ('2019-08-22',2,28)

insert into #payments\_interest values ('2019-08-23',2,23)

insert into #payments\_interest values ('2019-08-29',2,84)

# Приложение – таблицы в Excel



### Решение:

select i.agr\_id as agr\_id, p.payment\_sum as principal, i.payment\_sum as interest

from (select agr\_id, sum(payment\_sum) as payment\_sum

from #payments\_principal

group by agr\_id) p

inner join

(select agr\_id, sum(payment\_sum) as payment\_sum

from #payments\_interest

group by agr\_id) i

on p.agr\_id = i.agr\_id

### Ответ:

|  |  |  |
| --- | --- | --- |
| agr\_id | principal | interest |
| 1 | 35447 | 324 |
| 2 | 22953 | 212 |
| 3 | 19027 | 97 |
| 4 | 18340 | 110 |
| 5 | 16688 | 54 |
| 7 | 25654 | 270 |
| 9 | 8505 | 22 |
| 31 | 8281 | 98 |

## Задача №3

Написать скрипты на любом известном языке SQL, чтобы получить список кредитов, которые на момент расчета имеют непогашенную задолженность, и рассчитать для каждого такого кредита:

1. Общую (накопленную) сумму просроченного долга непогашенную (не выплаченную) к моменту расчета.
2. Дату начала текущей (последней) просрочки. Под датой начала просрочки, в данной задаче понимается первая дата непрерывного периода, в котором общая сумма просроченного непогашенного долга > 0.Учесть, что периодов просрочек может быть несколько.
3. Кол-во дней текущей просрочки.

**Исходные данные**

Таблица PDCL – содержит информацию о выносах на просрочку неоплаченных сумм по кредиту и о погашениях просроченного долга.

Структура

|  |  |
| --- | --- |
| Date | Дата |
| Customer | Номер клиента |
| Deal | Номер кредита |
| Currency | Валюта кредита |
| Sum | сумма, вынесенная на просрочку ("+") или выплаченная ("-") |

Пример:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Customer** | **Deal** | **Currency** | **Sum** |
| 12.12.2009 | 111110 | 111111 | RUR | 12000 |
| 25.12.2009 | 111110 | 111111 | RUR | 5000 |
| 12.12.2009 | 111110 | 122222 | RUR | 10000 |
| 12.01.2010 | 111110 | 111111 | RUR | -10100 |
| 20.11.2009 | 220000 | 222221 | RUR | 25000 |
| 20.12.2009 | 220000 | 222221 | RUR | 20000 |
| 31.12.2009 | 220001 | 222221 | RUR | -10000 |
| 29.12.2009 | 111110 | 122222 | RUR | -10000 |
| 27.11.2009 | 220001 | 222221 | RUR | -30000 |

Один клиент может иметь несколько кредитов. По кредиту выплаты могу совершать созаемщики. Если клиент, имеющий кредит, пропускает очередную выплату по графику, у него возникает просрочка. В таблице PDCL при этом появляется соответствующая запись, где Sum – невыплаченная сумма (с положительным знаком). Если, затем, клиент совершает выплату (полную сумму или ее часть) появляется новая запись, где Sum – выплаченная сумма (со знаком «-»). Следует учесть, что выплата клиента не обязательно полностью гасит накопленный долг, она может составлять лишь часть долга.

В приведенном примере по кредиту 111111 имеется невыплаченный просроченный долг на сумму 6900руб. (12000 + 5000 - 10100), дата начала текущей просрочки 12.12.2009.

**Скрипт**

USE [aero]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

select Deal, sum(Sum) as debt, min(Date) as dept\_start\_date, DATEDIFF(day, max(DATE), GETDATE()) as days\_of\_dapt

from PDCL

group by Deal

having sum(Sum)>0

GO



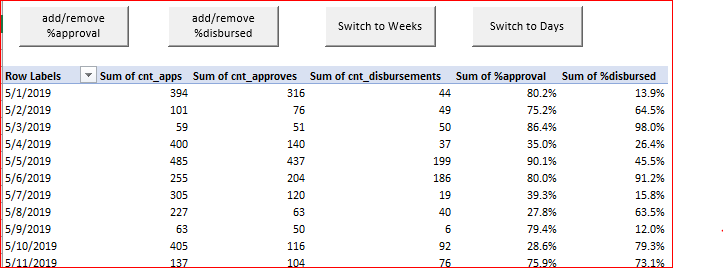
**Результат**

|  |  |  |  |
| --- | --- | --- | --- |
| Deal | debt | dept\_start\_date | days\_of\_dapt |
| 111111 | 6900 | 2009-12-12 | 4445 |
| 222221 | 5000 | 2009-11-20 | 4457 |

## Задача №4

Написать код на VBA для 4х кнопок (эксель файл с данными прилагается ниже):

* Add/remove %approval. По нажатию кнопки из сводной таблицы должно уйти поле «Sum of %approval» (если оно уже отображается). Если поле не выведено в сводную таблицу – оно должно снова туда добавиться.
* Add/remove %disbursed. Тоже самое, что и для %approved, только с полем «Sum of %disbursed».
* Switch to Weeks. По нажатию кнопки в сводную таблицу в строки должно подставиться поле «rep\_week».
* Switch to Days. По нажатию кнопки в своднyю таблицу в строки должно подставиться поле «rep\_date».



**Приложение:**



### Решение:

