

Тестовое задание от СБЕРА

Данные:

CLIENT – таблица с клиентами.

Содержит: ID клиента, ИНН, Наименование клиента.

HOLDING – таблица холдингов.

Содержит: Наименование холдинга, ИНН клиента, входящего в холдинг, дата начала (DB) и дата окончания (DE) принадлежности клиента к холдингу.

ZONES – зона благонадежности клиента.

Содержит: ИНН клиента, Зону (Zones), Тип регламента, по которому считалась зона (Type), статус карточки зоны активна/неактивна (ACTIVE), дату среза (DF), на которую зона была рассчитана.

CRED – реестр кредитов.

Содержит: ID клиента, Номер кредита (CRED), Остаток ссудной задолженности (OST), дату среза (DF), на которую остаток по кредиту был рассчитан.

Учтем необходимые ограничения к типам данных при импортировании неструктурированных данных, указав соответствующий тип данных:

- ИНН клиента состоит только из цифр, длина 10 или 12 символов. Тип поля при создании таблицы в базе данных всегда выбирать текстовый;
- Номер кредита состоит только из английских букв и/или цифр.

1 Задача.

Найти для каждого среза общую сумму остатка задолженности в разбивке по зонам

В ответе вывести зону, дату среза по возрастанию и остаток задолженности по убыванию.

При решении данной задачи необходимо учесть следующее:

- Дата среза говорит о том, что на эту (и только эту) дату произведен некий расчет
- Данные в базах хранятся с разных миграций и могут возникать дубли
- Если у клиента несколько зон, то берется, рассчитанная по более позднему регламенту
- Карточки со статусом неактивна не используются при расчетах

Предварительная подготовка данных **ДЛЯ РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ!!!**

Импортируем неструктурированные данные в формате SQL используя программный продукт Microsoft SQL Server Management Studio таблицы и учтём указанные ограничения:

SQLQuery13.sql - ...BS\ABezzubov (60))* SQLQuery12.sql - ...BS\ABezzubov (54))* SQLQuery7.sql - A...BS\ABezzubov (62))*

```
SELECT * FROM CLIENT;  
SELECT * FROM CRED;  
SELECT * FROM ZONES;
```

100 %

Результаты Сообщения

ID_CLIENT	INN	NAME
1227907787756	197553400466	PFCJKL
1309116181590	2879848400	IMUKTV
1504776973636	8580960492	VXKQOT
1916634235399	137175390694	RQRMGY
2342076343592	90039860887	DUDHJI
2645470240503	7123117752	GTIWUF
2837689832926	6705527674	VVDUIQ
2957703450963	226457514910	LDDJMQ

ID_CLIENT	CRED	DF	OST
7771756001883	14JHQ877	2021-01-01 ...	8222
5835613988348	8RY95QLS	2021-01-01 ...	5364
7029341434208	DQSQ5ECX	2021-01-01 ...	1899
1309116181590	5G4CYOVC	2021-01-01 ...	5980
4078053083534	5HKEU205H	2021-01-01 ...	13258
4061419452762	OCSUP90K	2021-01-01 ...	5431
3888849901185	4JYRQD4H	2021-01-01 ...	8022
6149484617433	9SGJOC67	2021-01-01 ...	8377

INN	DF	ZONES	TYPE	ACTIVE
2029812407	2021-01-01 ...	G	2020	Y
6133874085	2021-01-01 ...	G	2020	Y
7424446757	2021-01-01 ...	G	2020	Y
837934663463	2021-01-01 ...	B	2017	Y
4970580948	2021-01-01 ...	G	2017	Y
147860679878	2021-01-01 ...	G	2020	Y
6357833456	2021-01-01 ...	G	2020	Y
5787782049	2021-01-01 ...	G	2020	Y

Запрос успешно выполнен. ABEZZUBOV\SQLEXPRESS (14.0 ... IBS\ABezzubov (60) CLIENT 00:00:00 231 строки

```

WITH x AS (SELECT zn.INN, DF, ZONES ,TYPE,
MAX(TYPE) OVER (PARTITION BY zn.INN, DF ORDER BY DF DESC) as 'MAX_TYPE' FROM
ZONES as zn WHERE ACTIVE='Y'),
y AS (SELECT cl.INN, DF, OST FROM CRED as cr
JOIN CLIENT as cl ON cl.ID_CLIENT=cr.ID_CLIENT),
z AS (SELECT * FROM x WHERE TYPE=MAX_TYPE)
SELECT z.DF, ZONES, SUM(OST) AS 'REMAINING_DEP' FROM y
JOIN z ON z.DF=y.DF AND z.INN=y.INN
GROUP BY z.DF, ZONES
ORDER BY DF ASC;

```

```

WITH x AS (SELECT zn.INN, DF, ZONES ,TYPE,
MAX(TYPE) OVER (PARTITION BY zn.INN, DF ORDER BY DF DESC) as 'MAX_TYPE' FROM
ZONES as zn WHERE ACTIVE='Y'),
y AS (SELECT cl.INN, DF, OST FROM CRED as cr JOIN CLIENT as cl ON
cl.ID_CLIENT=cr.ID_CLIENT)
SELECT y.DF, SUM(y.OST) AS 'No ZONES!!!', x.ZONES FROM y
LEFT JOIN x ON y.DF=x.DF AND x.INN=y.INN GROUP BY y.DF, x.ZONES
HAVING x.ZONES IS NULL;

```

SQLQuery12.sql - ...BS\ABezzubov (54))* SQLQuery7.sql - A...BS\ABezzubov (62))*

```
WITH x AS (SELECT zn.INN, DF, ZONES ,TYPE,
MAX(TYPE) OVER (PARTITION BY zn.INN, DF ORDER BY DF DESC) as 'MAX_TYPE' FROM ZONES as zn WHERE ACTIVE='Y'),
y AS (SELECT cl.INN, DF, OST FROM CRED as cr
JOIN CLIENT as cl ON cl.ID_CLIENT=cr.ID_CLIENT),
z AS (SELECT * FROM x WHERE TYPE=MAX_TYPE)
SELECT z.DF, ZONES, SUM(OST) AS 'REMAINING_DEP' FROM y
JOIN z ON z.DF=y.DF AND z.INN=y.INN
GROUP BY z.DF, ZONES
ORDER BY DF ASC;

WITH x AS (SELECT zn.INN, DF, ZONES ,TYPE,
MAX(TYPE) OVER (PARTITION BY zn.INN, DF ORDER BY DF DESC) as 'MAX_TYPE' FROM ZONES as zn WHERE ACTIVE='Y'),
y AS (SELECT cl.INN, DF, OST FROM CRED as cr JOIN CLIENT as cl ON cl.ID_CLIENT=cr.ID_CLIENT)
SELECT y.DF, SUM(y.OST) AS 'No ZONES!!!', x.ZONES FROM y
LEFT JOIN x ON y.DF=x.DF AND x.INN=y.INN GROUP BY y.DF, x.ZONES
HAVING x.ZONES IS NULL;
```

100 %

Результаты Сообщения

	DF	ZONES	REMAINING_DEP
1	2021-01-01 00:00:00.0000000	B	28562
2	2021-01-01 00:00:00.0000000	G	312963
3	2021-01-01 00:00:00.0000000	N	112348
4	2021-04-01 00:00:00.0000000	B	45101
5	2021-04-01 00:00:00.0000000	G	240455
6	2021-04-01 00:00:00.0000000	N	123563
7	2021-07-01 00:00:00.0000000	B	36000
8	2021-07-01 00:00:00.0000000	G	274084
9	2021-07-01 00:00:00.0000000	N	84205

	DF	No ZONES!!!	ZONES
1	2021-01-01 00:00:00.0000000	35680	NULL
2	2021-04-01 00:00:00.0000000	27936	NULL
3	2021-07-01 00:00:00.0000000	19040	NULL

2 Задача.

Промежуточный отчет: рассчитать на каждый срез сумму остатка ссудной задолженности по следующему условию:

- Если клиент входит в холдинг, то вывести название холдинга и сумму остатка ссудной задолженности по холдингу
- Если клиент не входит в холдинг, то вывести ИНН клиента и сумму остатка ссудной задолженности по клиенту

В итоговом отчете вывести топ-10 по остатку задолженности клиента/холдинга, исходя из данных всех срезов. Отчет должен содержать ИНН клиента/наименование холдинга, дату среза, сумму остатка ссудной задолженности. В ответе предоставить только итоговый отчет.

- Данные в базах хранятся с разных миграций и могут возникать дубли

```
WITH client_holding AS (SELECT INN, HOLDING FROM HOLDING),
client_inn AS (SELECT cl.INN, DF, OST FROM CRED as cr
JOIN CLIENT as cl ON cl.ID_CLIENT=cr.ID_CLIENT),
holdings AS (SELECT HOLDING, SUM(OST) as 'Sum_OST' FROM client_holding
JOIN client_inn ON client_holding.INN=client_inn.INN GROUP BY HOLDING),
client_no_holding AS (SELECT client_inn.INN, client_holding.HOLDING,
client_inn.DF, SUM(OST) as 'Sum_OST' FROM client_inn
LEFT JOIN client_holding ON client_holding.INN=client_inn.INN WHERE
client_holding.HOLDING is NULL GROUP BY client_inn.INN, client_inn.DF
,client_holding.HOLDING),
cl_hol AS (SELECT client_holding.INN, client_holding.HOLDING, client_inn.DF,
SUM(client_inn.OST) as OSTs FROM client_holding
JOIN client_inn ON client_inn.INN=client_holding.INN
GROUP BY client_holding.INN, client_holding.HOLDING, client_inn.DF)
SELECT TOP(10) * FROM (SELECT * FROM cl_hol
UNION
SELECT * FROM client_no_holding) as a
ORDER BY OSTs DESC
```

Результаты		Сообщения		
	INN	HOLDING	DF	OSTs
1	459797763596	DYJJ	2021-04-01 00:00:00.0000000	72758
2	145363944441	NULL	2021-07-01 00:00:00.0000000	39776
3	5779709784	AMXY	2021-01-01 00:00:00.0000000	36000
4	459797763596	DYJJ	2021-07-01 00:00:00.0000000	35978
5	4691071097	NULL	2021-04-01 00:00:00.0000000	28674
6	18184587217	ZVDR	2021-07-01 00:00:00.0000000	23918
7	107428961174	WSSD	2021-07-01 00:00:00.0000000	20359
8	8164410539	JNIS	2021-07-01 00:00:00.0000000	20356
9	950868863368	NULL	2021-07-01 00:00:00.0000000	19648
10	137175390694	NULL	2021-01-01 00:00:00.0000000	19309

3 Задача* (со звёздочкой).

После начала проверочных действий от проверяемых подразделений поступило следующее сообщение:

«В реестре кредитов на 01.07.21 были найдены ошибки, высылаем актуальную версию». (файл ERR_REESTR.txt)

Мы, конечно, очень расстроились, так как уже сделали часть проверки на данных старого реестра, но глубоко вздохнув, нам всё же придётся переделать уже сделанную работу.

Из плохих новостей: реестр собирался впопыхах, дабы успеть к окончанию проверки, часть данных вносилась вручную, а значит велик риск операционных ошибок, в дополнение туда вошли кредиты, которые не проходили по счетам, а были созданы, для тестирования работы системы или не были выданы вовсе.

Из хороших новостей: мы точно знаем, что все те кредиты, которые были в старом реестре в том или ином виде есть в новом реестре.

Необходимо средствами python и/или SQL вывести все идентификаторы кредита (ID_CLIENT, CRED) из старого реестра и к ним подтянуть сумму остатка ссудной задолженности из нового, который прислали проверяемые подразделения. Также необходимо прислать код.