



лаб4

Создать 4 различных хранимых процедуры:

а) Процедура без параметров, формирующая список клиентов, арендующих на данный момент 2 и более торговые точки

```
1 CREATE PROCEDURE ClientsWithMultipleRental
2 AS
3 BEGIN
4     SELECT
5         c.ID,
6         c.Название,
7         c.ФИО_Контактного_Лица,
8         COUNT(cont.ID) AS КоличествоТочек
9     FROM
10    Client c
11   JOIN Contract cont ON c.ID = cont.ID_Клиента
12   JOIN RetailPoint rp ON cont.ID_Точки = rp.ID
13 WHERE
14     cont.Дата_Окончания >= GETDATE()
15     AND rp.Статус = 0
16 GROUP BY
17     c.ID, c.Название, c.ФИО_Контактного_Лица
18 HAVING
19     COUNT(cont.ID) >= 2
20 ORDER BY
21     КоличествоТочек DESC;
22 END;
23
24 EXEC ClientsWithMultipleRental;
```

ID	Название	ФИО_Контактного_Лица	КоличествоТочек
----	----------	----------------------	-----------------

б) Процедура, на входе получающая идентификатор торговой точки и формирующая список арендаторов, когда-либо использовавших эту точку

```
1 CREATE PROCEDURE GetPointRenters
2     @PointID INT
3 AS
4 BEGIN
5     SELECT
6         c.ID,
7         c.Название,
8         c.ФИО_Контактного_Лица,
9         cont.Номер_Договора,
10        cont.Дата_Заключения,
```

ID	Название	ФИО_Контактного_Лица	Номер_Договора	Дата_Заключения	Дата_Окончания	Финальная_Стоимость
1	ООО "Вкусные продукты"	Иванов Петр Сергеевич	ДГ-2024-001	2024-01-15	2024-12-31	25000.00

```

11     cont.Дата_Окончания,
12     cont.Финальная_Стоимость
13
14     Contract cont
15     JOIN Client c ON cont.ID_Клиента = c.ID
16
17     WHERE
18         cont.ID_Точки = @PointID
19
20 END;
21
20 EXEC GetPointRenters @PointID = 1;

```

с) Процедура, на входе получающая название(идентификатор) арендатора, выходной параметр – дата последнего договора, заключенного с ним, и адрес торговой точки, арендованной по этому договору

```

1 CREATE PROCEDURE GetLastContractInfo
2     @ClientID INT,
3     @LastContractDate DATE OUTPUT,
4     @PointAddress NVARCHAR(255) OUTPUT
5
6 AS
7 BEGIN
8     SELECT TOP 1
9         @LastContractDate = cont.Дата_Заключения,
10        @PointAddress = rp.Адрес
11
12     FROM
13         Contract cont
14     INNER JOIN RetailPoint rp ON cont.ID_Точки = rp.ID
15
16     WHERE
17         cont.ID_Клиента = @ClientID
18
19     ORDER BY
20         cont.Дата_Заключения DESC;
21
22 IF @LastContractDate IS NULL
23 BEGIN
24     SET @LastContractDate = NULL;
25     SET @PointAddress = 'Договоры не найдены';
26 END
27
28 END;
29
30
31 DECLARE @LastDate DATE, @Address NVARCHAR(255);
32 EXEC GetLastContractInfo
33     @ClientID = 1,
34     @LastContractDate = @LastDate OUTPUT,
35     @PointAddress = @Address OUTPUT;
36
37
38 SELECT
39     @LastDate AS ДатаПоследнегоДоговора,
40     @Address AS АдресТорговойТочки;

```

	ДатаПоследнегоДоговора	АдресТорговойТочки
1	2024-01-15	ул. Ленина, д. 15, ТЦ "Центральный"

d) Процедура, вызывающая вложенную процедуру, которая находит среднюю стоимость аренды торговой точки в месяц. Главная процедура для всех торговых точек, имеющих более низкую стоимость аренды и не находящихся в данный момент в аренде, увеличивает стоимость аренды на величину, равную разности между текущей стоимостью и средней

Возможно я не правильно поняла формулировку, но, если мы прибавляем к текущей стоимости разность между средней и текущей, то получается

$$T + (Сред-Т)=Сред.$$

Получается нужно просто присвоить этим точкам среднюю стоимость.

я добавила две свободные точки

```

1 INSERT INTO RetailPoint (Адрес,Этаж,Площадь,Наличие_Кондиционера,Стоимость_Аренды,Статус) VALUES
2     ('ул. Пушкина, 10, ТЦ "Центральный", 2, 45.50, 1, 25000.00, 1),
3     ('пр-т Ленина, 50, ТРК "МегаПлаза", 1-й этаж', 1, 60.00, 0, 35000.00, 1);

```

до

ID	Адрес	Этаж	Площадь	Наличие_Кондиционера	Стоимость_Аренды	Статус
1	ул. Ленина, д. 15, ТЦ "Центральный"	1	45.50	1	25000.00	0
2	пр. Мира, д. 28, ТЦ "Европа"	2	30.25	1	28250.00	0
3	ул. Советская, д. 5, ТЦ "Атриум"	1	60.00	1	35000.00	0
4	ул. Кирова, д. 12, ТЦ "Нева"	3	25.75	0	28250.00	0
5	пр. Победы, д. 45, ТЦ "Виктория"	1	80.20	1	45000.00	0
6	ул. Гагарина, д. 8, ТЦ "Орбита"	2	35.40	1	28250.00	0
7	ул. Пушкина, д. 33, ТЦ "Литера"	1	55.80	1	32000.00	0
8	ул. Садовая, д. 20, ТЦ "Флора"	2	40.30	0	28250.00	0
9	пр. Космонавтов, д. 17, ТЦ "Галактика"	1	70.15	1	38000.00	0
10	ул. Лесная, д. 3, ТЦ "Тайга"	3	28.90	1	28250.00	0
11	ул. Пушкина, 10, ТЦ "Центральный", 2-й этаж	2	45.50	1	25000.00	1
12	пр-т Ленина, 50, ТРК "МегаПлаза", 1-й этаж	1	60.00	0	35000.00	1

Средняя стоимость аренды: 31354.17

после

ID	Адрес	Этаж	Площадь	Наличие_Кондиционера	Стоимость_Аренды	Статус
1	ул. Ленина, д. 15, ТЦ "Центральный"	1	45.50	1	25000.00	0
2	пр. Мира, д. 28, ТЦ "Европа"	2	30.25	1	28250.00	0
3	ул. Советская, д. 5, ТЦ "Атриум"	1	60.00	1	35000.00	0
4	ул. Кирова, д. 12, ТЦ "Нева"	3	25.75	0	28250.00	0
5	пр. Победы, д. 45, ТЦ "Виктория"	1	80.20	1	45000.00	0
6	ул. Гагарина, д. 8, ТЦ "Орбита"	2	35.40	1	28250.00	0
7	ул. Пушкина, д. 33, ТЦ "Литера"	1	55.80	1	32000.00	0
8	ул. Садовая, д. 20, ТЦ "Флора"	2	40.30	0	28250.00	0
9	пр. Космонавтов, д. 17, ТЦ "Галактика"	1	70.15	1	38000.00	0
10	ул. Лесная, д. 3, ТЦ "Тайга"	3	28.90	1	28250.00	0
11	ул. Пушкина, 10, ТЦ "Центральный", 2-й этаж	2	45.50	1	31354.17	1
12	пр-т Ленина, 50, ТРК "МегаПлаза", 1-й этаж	1	60.00	0	35000.00	1

изменилась 11 точка

3 пользовательских функции:

a) Скалярная функция, которая для заданной точки подсчитывает выручку от ее аренды за текущий год

```

1 CREATE FUNCTION dbo.GetPointRevenue (
2     @PointID INT
3 )
4 RETURNS DECIMAL(18, 2)
5 AS
6 BEGIN
7     DECLARE @Revenue DECIMAL(18, 2) = 0;
8
9     SELECT @Revenue = SUM(p.Сумма)
10    FROM Payment p
11   JOIN Contract c ON p.ID_Договора = c.ID
12   WHERE
13       c.ID_Точки = @PointID
14      AND YEAR(p.Дата_Оплаты) = 2024 --я поставила 2024 вместо YEAR(GETDATE()), т.к. у меня нет платежей за
15      2025
16      AND p.Статус = 1;
17
18    RETURN ISNULL(@Revenue, 0);
19 END;
19 SELECT dbo.GetPointRevenue(5) AS Выручка_За_Год;

```

	Выручка_За_Год
1	225000.00

b) Inline-функция, возвращающая идентификаторы торговых точек, еще не сдававшихся в аренду в текущем году

```

1 CREATE FUNCTION dbo.GetUnrentedPointsThisYear ()
2 RETURNS TABLE
3 AS
4 RETURN (
5     SELECT rp.ID, rp.Адрес
6     FROM RetailPoint rp
7     WHERE NOT EXISTS (
8         SELECT 1
9         FROM Contract c
10        WHERE
11            c.ID_Точки = rp.ID
12            AND YEAR(c.Дата_Заключения) <= 2024
13            AND YEAR(c.Дата_Окончания) >= 2024 --я поставила 2024 вместо YEAR(GETDATE()), т.к. у меня нет
14            договоров за 2025
15    );
16 SELECT * FROM dbo.GetUnrentedPointsThisYear();

```

	ID	Адрес
1	11	ул. Пушкина, 10, ТЦ "Центральный", 2-й этаж
2	12	пр-т Ленина, 50, ТРК "МегаПлаза", 1-й этаж

c) Multi-statement-функция, выдающая список арендаторов, которые когда-либо задерживали ежемесячную оплату в виде: Название арендатора, номер договора, адрес торговой точки, год и месяц оплаты

(Ежемесячная оплата за аренду считается задержанной, если она поступает позднее 10-го числа месяца, следующего за оплачиваемым)

```

1 CREATE FUNCTION dbo.GetLatePayers ()
2 RETURNS @LatePayments TABLE (
3     Арендатор NVARCHAR(255),
4     Номер_Договора NVARCHAR(50),
5     Адрес_Точки NVARCHAR(255),

```

```

6 Год_Оплаты INT,
7 Месяц_Оплаты INT,
8 Дата_Оплаты DATE
9 )
10 AS
11 BEGIN
12   INSERT INTO @LatePayments
13   SELECT
14     cl.Название,
15     c.Номер_Договора,
16     rp.Адрес,
17     YEAR(p.Дата_Оплаты) AS Год_Оплаты,
18     p.Месяц AS Месяц_Оплаты,
19     p.Дата_Оплаты
20   FROM Payment p
21   JOIN Contract c ON p.ID_Договора = c.ID
22   JOIN Client cl ON c.ID_Клиента = cl.ID
23   JOIN RetailPoint rp ON c.ID_Точки = rp.ID
24   WHERE
25     p.Статус = 1
26     AND DAY(p.Дата_Оплаты) > 10;
27   RETURN;
28 END;
29 SELECT * FROM dbo.GetLatePayers();

```

	Арендатор	Номер_Договора	Адрес_Точки	Год_Оплаты	Месяц_Оплаты	Дата_Оплаты
1	ООО "Вкусные продукты"	ДГ-2024-001	ул. Ленина, д. 15, ТЦ "Центральный"	2024	1	2024-01-15
2	ООО "Вкусные продукты"	ДГ-2024-001	ул. Ленина, д. 15, ТЦ "Центральный"	2024	2	2024-02-15
3	ООО "Вкусные продукты"	ДГ-2024-001	ул. Ленина, д. 15, ТЦ "Центральный"	2024	3	2024-03-15
4	ООО "Вкусные продукты"	ДГ-2024-001	ул. Ленина, д. 15, ТЦ "Центральный"	2024	4	2024-04-15
5	ООО "Вкусные продукты"	ДГ-2024-001	ул. Ленина, д. 15, ТЦ "Центральный"	2024	5	2024-05-15
6	ООО "Вкусные продукты"	ДГ-2024-001	ул. Ленина, д. 15, ТЦ "Центральный"	2024	6	2024-06-15
7	ООО "Вкусные продукты"	ДГ-2024-001	ул. Ленина, д. 15, ТЦ "Центральный"	2024	7	2024-07-15
8	ООО "Модный стиль"	ДГ-2024-003	ул. Советская, д. 5, ТЦ "Атриум"	2024	1	2024-01-20
9	ООО "Модный стиль"	ДГ-2024-003	ул. Советская, д. 5, ТЦ "Атриум"	2024	2	2024-02-20
10	ООО "Модный стиль"	ДГ-2024-003	ул. Советская, д. 5, ТЦ "Атриум"	2024	3	2024-03-20
11	ООО "Модный стиль"	ДГ-2024-003	ул. Советская, д. 5, ТЦ "Атриум"	2024	4	2024-04-20
12	ООО "Модный стиль"	ДГ-2024-003	ул. Советская, д. 5, ТЦ "Атриум"	2024	5	2024-05-20
13	ООО "Модный стиль"	ДГ-2024-003	ул. Советская, д. 5, ТЦ "Атриум"	2024	6	2024-06-20
14	ООО "Книжный мир"	ДГ-2024-005	пр. Победы, д. 45, ТЦ "Виктория"	2024	1	2024-02-15
15	ООО "Книжный мир"	ДГ-2024-005	пр. Победы, д. 45, ТЦ "Виктория"	2024	2	2024-03-15
16	ООО "Книжный мир"	ДГ-2024-005	пр. Победы, д. 45, ТЦ "Виктория"	2024	3	2024-04-15
17	ООО "Книжный мир"	ДГ-2024-005	пр. Победы, д. 45, ТЦ "Виктория"	2024	4	2024-05-15
18	ООО "Книжный мир"	ДГ-2024-005	пр. Победы, д. 45, ТЦ "Виктория"	2024	5	2024-06-15

Создать 3 триггера:

a) Триггер любого типа на добавление договора аренды – если клиент хотя бы 1 раз ранее задерживал оплату, то сумма договора для него увеличивается на 3%, если он задерживал оплату 2 и более раз – договор не заключается

```

1 CREATE TRIGGER trg_InsertContract
2 ON Contract
3 INSTEAD OF INSERT
4 AS
5 BEGIN
6
7   DECLARE @LatePayers TABLE (
8     ClientID INT,
9     LateCount INT
10 );
11
12
13   INSERT INTO @LatePayers (ClientID, LateCount)
14   SELECT
15     i.ID_Клиента,
16     COUNT(*) AS LateCount
17   FROM inserted i
18   JOIN Payment p ON i.ID_Клиента = p.ID_Договора
19   WHERE
20     p.Статус = 1
21     AND DAY(p.Дата_Оплаты) > 10
22   GROUP BY i.ID_Клиента;

```

```

1 INSERT INTO Contract (
2   Номер_Договора, Дата_Заключения, Дата_Окончания,
3   Финальная_Стоимость, ID_Клиента, ID_Точки, ID_Скидки
4 )
5 VALUES (
6   'ДГ-2024-011', '2024-04-01', '2024-12-31',
7   50000.00, 1, 1, 1
8 );

```

сообщение: 50001, уровень: 16, состояние: 1, процедура: trg_InsertContract, строка: 30 [строка начала пакета: 0]
Договор не может быть заключен: клиент имеет 2 и более просроченных платежа.

```

23
24 IF EXISTS (
25   SELECT 1
26   FROM @LatePayers
27   WHERE LateCount >= 2
28 )
29 BEGIN
30   THROW 50001, 'Договор не может быть заключён: клиент имеет 2 и более просроченных
платежа.', 1;
31   RETURN;
32 END
33
34
35 INSERT INTO Contract
36   (Номер_Договора, Дата_Заключения, Дата_Окончания, Финальная_Стоимость, ID_Клиента, ID_Точки, ID_Скидки)
37   SELECT
38     i.Номер_Договора, i.Дата_Заключения, i.Дата_Окончания,
39     CASE
40       WHEN l.LateCount = 1 THEN i.Финальная_Стоимость * 1.03
41
42       ELSE i.Финальная_Стоимость
43     END AS Финальная_Стоимость,
44     i.ID_Клиента, i.ID_Точки, i.ID_Скидки
45   FROM inserted i
46   LEFT JOIN @LatePayers l ON i.ID_Клиента = l.ClientID;
47 END;

```

б) Последующий триггер на изменение стоимости аренды торговой точки – стоимость торговой точки не может быть увеличена более, чем на разность между текущей стоимостью аренды точки и средней стоимостью аренды по всем точкам

```

1
2 CREATE TRIGGER trg_UpdateRentalCost
3 ON RetailPoint
4 AFTER UPDATE
5 AS
6 BEGIN
7   DECLARE @Updates TABLE (
8     ID INT,
9     OldCost DECIMAL(12, 2),
10    NewCost DECIMAL(12, 2)
11  );
12
13  INSERT INTO @Updates (ID, OldCost, NewCost)
14  SELECT
15    i.ID, d.Стоимость_Аренды AS OldCost, i.Стоимость_Аренды AS NewCost
16  FROM inserted i
17  JOIN deleted d ON i.ID = d.ID
18  WHERE i.Стоимость_Аренды <> d.Стоимость_Аренды;
19
20  DECLARE @AvgCost DECIMAL(12, 2);
21  SELECT @AvgCost = AVG(Стоимость_Аренды) FROM RetailPoint;
22  IF EXISTS (
23    SELECT 1
24    FROM @Updates u
25    WHERE u.NewCost - u.OldCost > ABS(@AvgCost - u.OldCost)
26  )
27  BEGIN
28    THROW 50002, 'Ошибка: стоимость аренды не может быть увеличена более чем до средней стоимости по
всем точкам.', 1;
29    RETURN;
30  END
31
32 END;

```

было

ID	Адрес	Этаж	Площадь	Наличие_Кондиционера	Стоимость_Аренды	Статус
1	ул. Ленина, д. 15, ТЦ "Центральный"	1	45.50	1	25000.00	0
2	пр. Мира, д. 28, ТЦ "Европа"	2	30.25	1	28250.00	0
3	ул. Советская, д. 5, ТЦ "Атриум"	1	60.00	1	35000.00	0
4	ул. Кирова, д. 12, ТЦ "Нева"	3	25.75	0	28250.00	0
5	пр. Победы, д. 45, ТЦ "Виктория"	1	80.20	1	45000.00	0
6	ул. Гагарина, д. 8, ТЦ "Орбита"	2	35.40	1	28250.00	0
7	ул. Пушкина, д. 33, ТЦ "Литера"	1	55.80	1	32000.00	0
8	ул. Садовая, д. 20, ТЦ "Флора"	2	40.30	0	28250.00	0
9	пр. Космонавтов, д. 17, ТЦ "Галакт..."	1	70.15	1	38000.00	0
10	ул. Лесная, д. 3, ТЦ "Тайга"	3	28.90	1	28250.00	0
11	ул. Пушкина, 10, ТЦ "Центральный..."	2	45.50	1	31354.17	1
12	пр-т Ленина, 50, ТРК "МегаПлаза"...	1	60.00	0	35000.00	1

1 UPDATE RetailPoint SET Стоимость_Аренды = 60000 WHERE ID = 1;

c) Замещающий триггер на операцию удаления – торговую точку можно удалить из каталога только в случае, если ее ни разу не арендовали, в противном случае – выдать соотв.сообщение и ничего не удалять.

Обязательно предусмотреть обработку НЕСКОЛЬКИХ записей! (там, где необходимо, использовать КУРСОР!)

```
1 CREATE TRIGGER trg_PreventDeleteRentedPoint
2 ON RetailPoint
3 INSTEAD OF DELETE
4 AS
5 BEGIN
6     IF EXISTS (
7         SELECT 1
8         FROM deleted d
9         JOIN Contract c ON d.ID = c.ID_Точки
10    )
11    BEGIN
12        THROW 50003,
13        'Ошибка: нельзя удалить торговую точку, которая была арендована.',1;
14        RETURN;
15    END
16    DELETE FROM RetailPoint
17    WHERE ID IN (SELECT ID FROM deleted);
18 END;
```

```
1 DELETE FROM RetailPoint WHERE ID = 10;
```

