

Лабораторная №4

1. Создать 4 различных хранимых процедуры:

а) Процедура без параметров, формирующая список вызовов по каждому врачу на текущий день

	id	appeal_date	start_time	end_time	visit_status	patient_id	therapist_id
	1	2025-11-01	09:00:00	09:30:00	True	1	1
	2	2025-11-01	10:00:00	10:30:00	True	3	2
	3	2025-11-01	11:00:00	11:30:00	True	5	3
▶	5	2024-01-17	09:30:00	10:00:00	True	9	4

```
1 CREATE PROCEDURE GetTodayCallsTest2
2 AS
3 BEGIN
4     SELECT
5         t.full_name AS therapist_name,
6         p.full_name AS patient_name,
7         c.start_time,
8         c.visit_status
9     FROM Call c
10    JOIN Therapist t ON c.therapist_id = t.id
11    JOIN Patient p ON c.patient_id = p.id
12    WHERE c.appeal_date = CAST(GETDATE() AS DATE)
13    ORDER BY t.full_name, c.start_time;
14 END;
15 GO
16 EXEC GetTodayCallsTest2;
```

100 % Проблемы не найдены.

Результаты Сообщения

	therapist_name	patient_name	start_time	visit_status
1	Вихарев Кирилл Алексеевич	Быков Александр Владимирович	10:00:00.0000000	1
2	Семенов Евгений Николаевич	Воробьева Мария Павловна	11:00:00.0000000	1
3	Шебета Николай Иванович	Андрианова Анастасия Дмитриевна	09:00:00.0000000	1

b) Процедура, на входе получающая номер участка и формирующая список улиц, находящихся на этом участке

The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows the definition of a stored procedure named `GetDistrictStreets` in the `dbo` schema. The procedure takes an integer parameter `@district_number` and returns a list of streets from the `District` table where the `district_number` matches the input. The bottom pane shows the results of executing the procedure with `@district_number = 1`. The results are displayed in a table with one column, `streets`, and one row containing the text "Улица Ленина, Улица Мира, Улица Гагарина".

```
1 GO
2 CREATE PROCEDURE GetDistrictStreets
3     @district_number INT
4 AS
5 BEGIN
6     SELECT streets
7     FROM District
8     WHERE district_number = @district_number;
9 END;
10 GO
11
12 EXEC GetDistrictStreets @district_number = 1;
```

100 % Проблемы не найдены.

Результаты Сообщения

	streets
1	Улица Ленина, Улица Мира, Улица Гагарина

с) Процедура, на входе получающая номер участка, как выходной параметр выдает ФИО врача, обслуживающего данный участок

```
1 CREATE PROCEDURE GetDistrictTherapist
2     @district_number INT,
3     @therapist_name NVARCHAR(100) OUTPUT
4 AS
5 BEGIN
6     SELECT @therapist_name = t.full_name
7     FROM Therapist t
8     INNER JOIN District d ON t.district_id = d.id
9     WHERE d.district_number = @district_number;
10 END;
11 GO
12
13 DECLARE @doctor NVARCHAR(100);
14 EXEC GetDistrictTherapist @district_number = 1, @therapist_name = @doctor OUTPUT;
15 SELECT @doctor AS doctor;
```

100 % Проблемы не найдены.

Результаты Сообщения

	doctor
1	Вихарев Кирилл Алексеевич

d) Процедура, находящаяся один из участков с максимальным количеством домов и возвращающая ФИО врача, обслуживающего данный участок (с использованием вызова предыдущей процедуры)

```
1 CREATE PROCEDURE GetTherapistMaxHouses
2     @therapist_name NVARCHAR(100) OUTPUT
3 AS
4 BEGIN
5     DECLARE @max_district INT;
6     SELECT TOP 1 @max_district = p.district_id
7     FROM Patient p
8     GROUP BY p.district_id
9     ORDER BY COUNT(DISTINCT p.house_number) DESC;
10    SELECT @therapist_name = t.full_name
11    FROM Therapist t
12    WHERE t.district_id = @max_district;
13 END;
14 GO
15 DECLARE @doctor_max NVARCHAR(100);
16 EXEC GetTherapistMaxHouses @therapist_name = @doctor_max OUTPUT;
17 SELECT @doctor_max AS doctor_max_houses;
```

100 % Проблемы не найдены.

Результаты Сообщения

	doctor_max_houses
1	Шебета Николай Иванович

2.3 пользовательских функции:

а) Скалярная функция, возвращающая по адресу (улица, дом) номер участка

```
1 CREATE FUNCTION GetDistrictByAddress(  
2     @street NVARCHAR(100),  
3     @house_number INT  
4 )  
5 RETURNS INT  
6 AS  
7 BEGIN  
8     DECLARE @district_number INT;  
9  
10    SELECT @district_number = d.district_number  
11    FROM Patient p  
12    INNER JOIN District d ON p.district_id = d.id  
13    WHERE p.street = @street AND p.house_number = @house_number;  
14  
15    RETURN @district_number;  
16 END;  
17 GO  
18  
19 SELECT dbo.GetDistrictByAddress(N'Улица Ленина', 10) as district_number;
```

100 % Проблемы не найдены.

Результаты Сообщения

	district_number
1	1

b) Inline-функция, возвращающая все вызовы, поступившие от заданного пациента за текущий год

```
1 CREATE FUNCTION GetPatientCallsCurrentYear(  
2     @patient_id INT  
3 )  
4 RETURNS TABLE  
5 AS  
6 RETURN  
7     SELECT  
8         appeal_date,  
9         start_time,  
10        end_time,  
11        visit_status  
12     FROM Call  
13     WHERE patient_id = @patient_id  
14           AND YEAR(appeal_date) = YEAR(GETDATE());  
15 GO  
16  
17 SELECT * FROM GetPatientCallsCurrentYear(1);
```

В базе данных уже есть объект с именем "GetPatientCallsCurrentYear".

100 % 1 0

Результаты Сообщения

	appeal_date	start_time	end_time	visit_status
1	2025-10-29	09:00:00.0000000	09:30:00.0000000	1

с) Multi-statement-функция, возвращающая для заданного терапевта список пациентов, ожидающих его посещения в формате: ФИО пациента, адрес, примерное время посещения (из расчета, что на каждое посещение врач тратит 30 мин.)

```
1 CREATE FUNCTION GetWaitingPatients(  
2     @therapist_id INT  
3 )  
4 RETURNS @result TABLE (  
5     patient_name NVARCHAR(100),  
6     address NVARCHAR(200),  
7     visit_time TIME  
8 )  
9 AS  
10 BEGIN  
11     INSERT INTO @result  
12     SELECT  
13         p.full_name,  
14         p.street + ' ' + CAST(p.house_number AS NVARCHAR(10)),  
15         c.start_time  
16     FROM Call c  
17     JOIN Patient p ON c.patient_id = p.id  
18     WHERE c.therapist_id = @therapist_id  
19         AND c.visit_status = 0;  
20  
21     RETURN;  
22 END;  
23 GO  
24 SELECT * FROM dbo.GetWaitingPatients(4);
```

100 % 1 0

Результаты Сообщения

	patient_name	address	visit_time
1	Чернышова Мария Ивановна	Улица Чехова 21	10:00:00.0000000

3. Создать 3 триггера:

а) Триггер любого типа на добавление нового врача – если номер участка совпадает с номером какого-то другого врача, то вывод предупреждения и запись не добавляется

```
1 CREATE TRIGGER PreventDuplicateDistrictTherapist
2 ON Therapist
3 INSTEAD OF INSERT
4 AS
5 BEGIN
6     IF EXISTS (
7         SELECT 1
8         FROM inserted i
9         INNER JOIN Therapist t ON i.district_id = t.district_id
10    )
11    BEGIN
12        PRINT 'Ошибка: Врач с таким номером участка уже существует!';
13        RETURN;
14    END
15
16    INSERT INTO Therapist (full_name, phone, office_number, district_id)
17    SELECT full_name, phone, office_number, district_id
18    FROM inserted;
19 END;
20 GO
21
22 INSERT INTO Therapist (full_name, phone, office_number, district_id)
23 VALUES ('Новый врач', '+7-900-000-00-00', 105, 1);
```

Ошибка: Врач с таким номером участка уже существует!

(Затронута 1 строка)

Время выполнения: 2025-10-29T22:50:39.0196146+03:00

b) Последующий триггер на изменение диагноза посещения – если диагноз уже был выставлен, то его менять нельзя

```
1 CREATE TRIGGER PreventDiagnosisUpdate
2 ON Call_diagnosis
3 FOR UPDATE
4 AS
5 BEGIN
6     IF EXISTS (SELECT 1 FROM inserted)
7     BEGIN
8         RAISERROR('Нельзя изменять уже выставленный диагноз!', 16, 1);
9         ROLLBACK TRANSACTION;
10    END
11 END;
12 GO
13
14 UPDATE Call_diagnosis
15 SET doctor_comment = 'Новый комментарий'
16 WHERE diagnosis_id = 1 AND call_id = 1;
```

00 % 1 0

Сообщения

сообщение: 50000, уровень: 16, состояние: 1, процедура: PreventDiagnosisUpdate, строка: 8 [строка начала пакета: 13]
Нельзя изменять уже выставленный диагноз!

Сообщение 3609, уровень 16, состояние 1, строка 14
Транзакция завершилась в триггере. Выполнение пакета прервано.

Время выполнения: 2025-11-01T15:50:37.1457750+03:00

с) Замещающий триггер на операцию отмены вызова – если врач еще не посетил этого пациента, то вызов можно отменить, в противном случае строка с этим вызовом не удаляется

	id	appeal_date	start_time	end_time	visit_status	patient_id	therapist_id
1	1	2025-10-29	09:00:00.0000000	09:30:00.0000000	1	1	1
2	2	2025-10-29	10:00:00.0000000	10:30:00.0000000	1	3	2
3	3	2025-10-29	11:00:00.0000000	11:30:00.0000000	1	5	3
4	4	2025-10-29	14:00:00.0000000	14:30:00.0000000	0	7	1
5	5	2024-01-17	09:30:00.0000000	10:00:00.0000000	1	9	4
6	6	2024-01-17	13:00:00.0000000	13:30:00.0000000	1	11	1
7	7	2024-01-18	10:30:00.0000000	11:00:00.0000000	1	13	2
8	8	2024-01-18	15:00:00.0000000	15:30:00.0000000	0	15	3
9	9	2024-01-19	11:30:00.0000000	12:00:00.0000000	1	17	4
10	10	2024-01-19	14:30:00.0000000	15:00:00.0000000	1	19	1
11	11	2024-01-22	09:00:00.0000000	09:30:00.0000000	1	21	2
12	12	2024-01-22	12:00:00.0000000	12:30:00.0000000	1	23	3
13	13	2024-01-23	10:00:00.0000000	10:30:00.0000000	0	25	4
14	14	2024-01-23	13:30:00.0000000	14:00:00.0000000	1	27	1
15	29	2024-01-24	09:00:00.0000000	09:30:00.0000000	1	2	1
16	30	2024-01-24	10:00:00.0000000	10:30:00.0000000	1	4	2
17	31	2024-01-24	11:00:00.0000000	11:30:00.0000000	1	6	3
18	32	2024-01-25	14:00:00.0000000	14:30:00.0000000	1	8	4
19	33	2024-01-25	15:00:00.0000000	15:30:00.0000000	1	10	1

```
1 CREATE VIEW CallView
2 AS
3 SELECT id, appeal_date, start_time, end_time, visit_status, patient_id, therapist_id
4 FROM Call;
5 GO
6
7 CREATE TRIGGER CancelCallViewTrigger
8 ON CallView
9 INSTEAD OF DELETE
10 AS
11 BEGIN
12     DELETE FROM Call
13     WHERE id IN (SELECT id FROM deleted WHERE visit_status = 0);
14     IF EXISTS (SELECT 1 FROM deleted WHERE visit_status = 1)
15     BEGIN
16         PRINT 'Нельзя отменить завершённые вызовы!';
17     END
18 END;
19 GO
20
21 DELETE FROM CallView WHERE id = 4;
22 DELETE FROM CallView WHERE id = 1;
```

100 % 1 0

Сообщения

(Затронута 1 строка)

(Затронута 1 строка)

```
-- || | DELETE FROM CallView WHERE id = 1;
```

0 % 1 0

Сообщения

(Затронута строк: 0)

Нельзя отменить завершённые вызовы!

(Затронута 1 строка)

Результаты Сообщения

	id	appeal_date	start_time	end_time	visit_status	patient_id	therapist_id
1	1	2025-10-29	09:00:00.0000000	09:30:00.0000000	1	1	1
2	2	2025-10-29	10:00:00.0000000	10:30:00.0000000	1	3	2
3	3	2025-10-29	11:00:00.0000000	11:30:00.0000000	1	5	3
4	5	2024-01-17	09:30:00.0000000	10:00:00.0000000	1	9	4
5	6	2024-01-17	13:00:00.0000000	13:30:00.0000000	1	11	1
6	7	2024-01-18	10:30:00.0000000	11:00:00.0000000	1	13	2
7	8	2024-01-18	15:00:00.0000000	15:30:00.0000000	0	15	3
8	9	2024-01-19	11:30:00.0000000	12:00:00.0000000	1	17	4
9	10	2024-01-19	14:30:00.0000000	15:00:00.0000000	1	19	1
10	11	2024-01-22	09:00:00.0000000	09:30:00.0000000	1	21	2
11	12	2024-01-22	12:00:00.0000000	12:30:00.0000000	1	23	3
12	13	2024-01-23	10:00:00.0000000	10:30:00.0000000	0	25	4
13	14	2024-01-23	13:30:00.0000000	14:00:00.0000000	1	27	1
14	29	2024-01-24	09:00:00.0000000	09:30:00.0000000	1	2	1
15	30	2024-01-24	10:00:00.0000000	10:30:00.0000000	1	4	2
16	31	2024-01-24	11:00:00.0000000	11:30:00.0000000	1	6	3
17	32	2024-01-25	14:00:00.0000000	14:30:00.0000000	1	8	4
18	33	2024-01-25	15:00:00.0000000	15:30:00.0000000	1	10	1