

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования «Ивановский государственный энергетический
университет имени В.И. Ленина».

Кафедра систем управления

Отчет по лабораторному практикуму
**«Проектирование баз данных и разработка приложений в СУБД
Microsoft SQL Server»**

Дисциплина: Системы управления базами данных

Вариант 10

Выполнили: студенты группы 2-43

Балашов И.

Ковшов Д.

Проверил: ассистент

Трифонов А. Ю.

Цель работы

Разработка и реализация базы данных для конкретной предметной области с использованием методов моделирования, таких как создание концептуальной, логической и физической моделей. В рамках работы необходимо определить таблицы пользовательских ролей и их функции, а также установить ограничения на вводимые данные, значения по умолчанию и правила удаления. Также требуется разработать и выполнить запросы на модификацию данных (Insert, Update, Delete) и запросы на выборку с использованием различных операторов и функций SQL, включая агрегацию, подзапросы и объединения. В процессе работы также предстоит создать хранимые процедуры и триггеры для автоматизации обработки данных и обеспечения целостности базы данных.

Задание

Оператор сотовой связи

Оператор сотовой связи МТС предоставляет услуги сотовой связи физическим и юридическим лицам по всей России. Юридические лица могут подключать только корпоративные тарифы, а физические – только некорпоративные. Для каждого клиента известно количество средств на счету. Для каждого тарифа указаны стоимость перехода на тариф, стоимость минуты разговоров (по городу, для междугородних и международных звонков). Все звонки фиксируются: кто звонит, кому звонят, время звонка, продолжительность разговора (целое количество минут), тип соединения (по городу, междугородний, международный). После окончания разговора со счета звонящего списывается определенная сумма. Каждый клиент вправе запросить распечатку звонков за любой период.

Проектирование

Создание модели предметной области начинается с определения абстракций, существующих в реальном мире, то есть ключевых концептуальных объектов, присутствующих в системе. Концептуальная модель представляет собой описание понятий через призму предметной области. Она отражает структуру проектируемой системы на высоком уровне абстракции, не привязанном к её физической реализации. При построении концептуальной модели предпочтительнее излишняя детализация, чем недостаточная проработка, поэтому в неё включаются все понятия предметной области, даже те, которые не планируется использовать при разработке базы данных.

1) Создание диаграммы классов (рис. 1)

В UML концептуальная модель изображается в виде статической диаграммы классов без указания операций. Такая диаграмма описывает типы объектов системы и различные виды статических связей между ними.

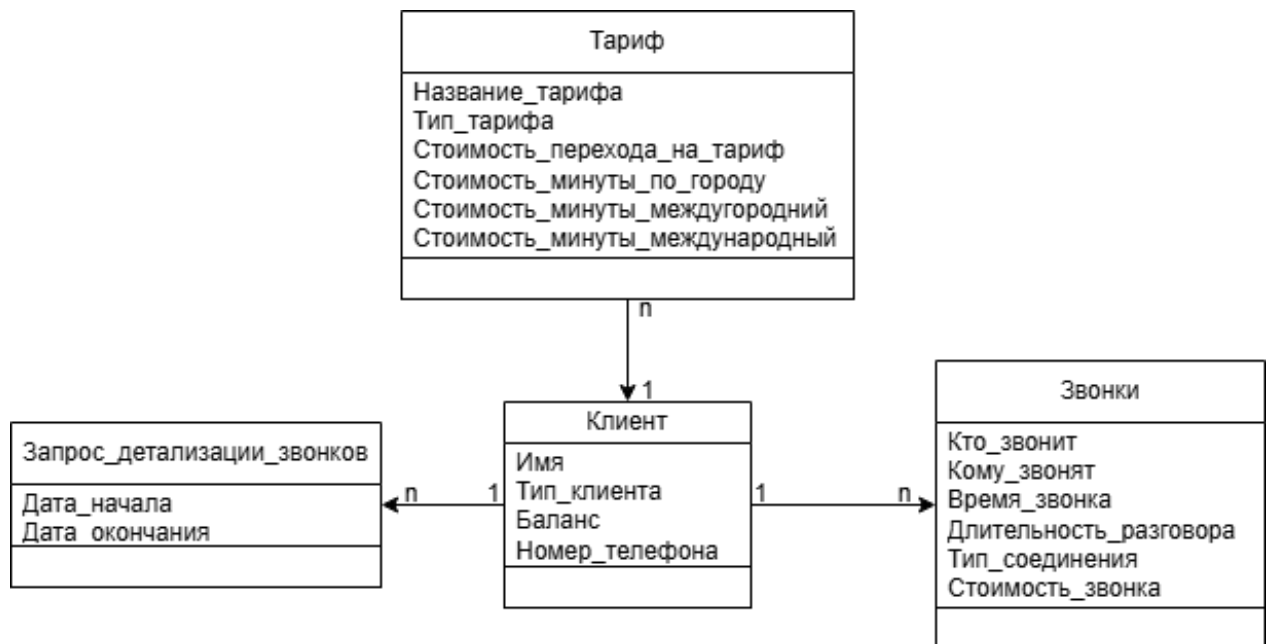


Рис. 1. Диаграмма классов.

2) Создание концептуальной модели базы данных (рис. 2)

Концептуальная модель базы данных описывает объекты предметной области, их атрибуты и взаимосвязи между ними в той степени, в которой они подлежат непосредственному сохранению в базе данных системы.

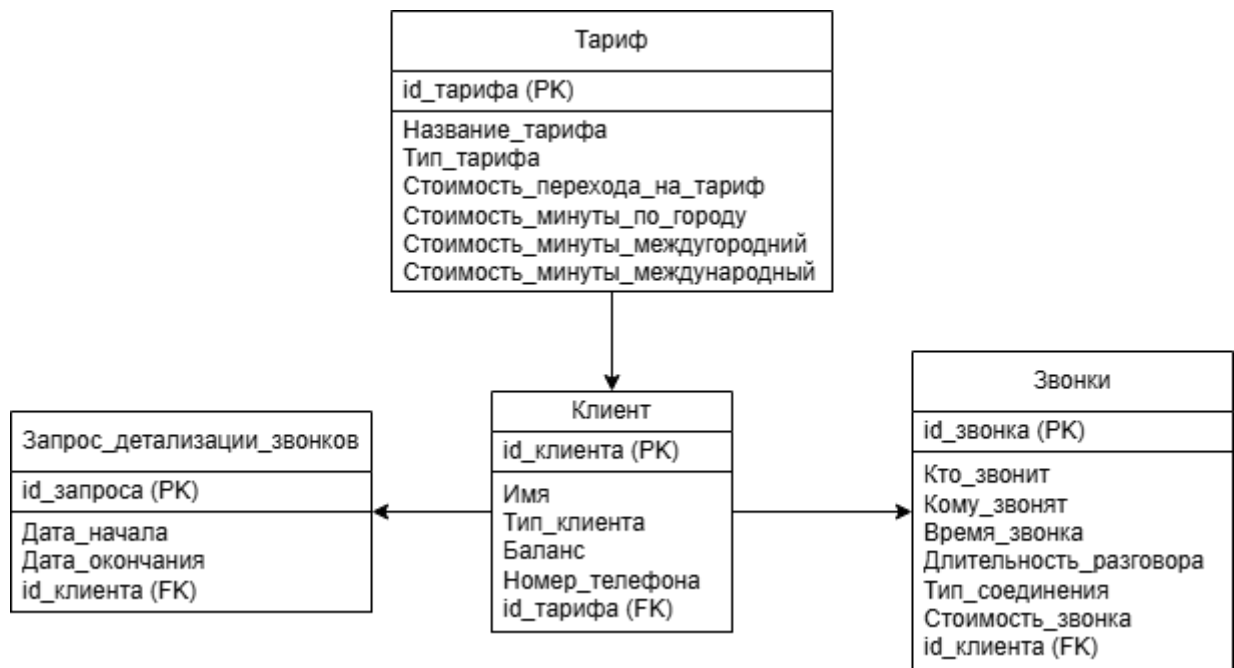


Рис. 2. Концептуальная модель базы данных

3) Создание физической модели базы данных (рис. 3)

Физическая модель включает в себя все необходимые детали для конкретной СУБД, обеспечивающие эффективное определение данных, такие как наименования столбцов, типы данных, описание первичных и внешних ключей и другие элементы.

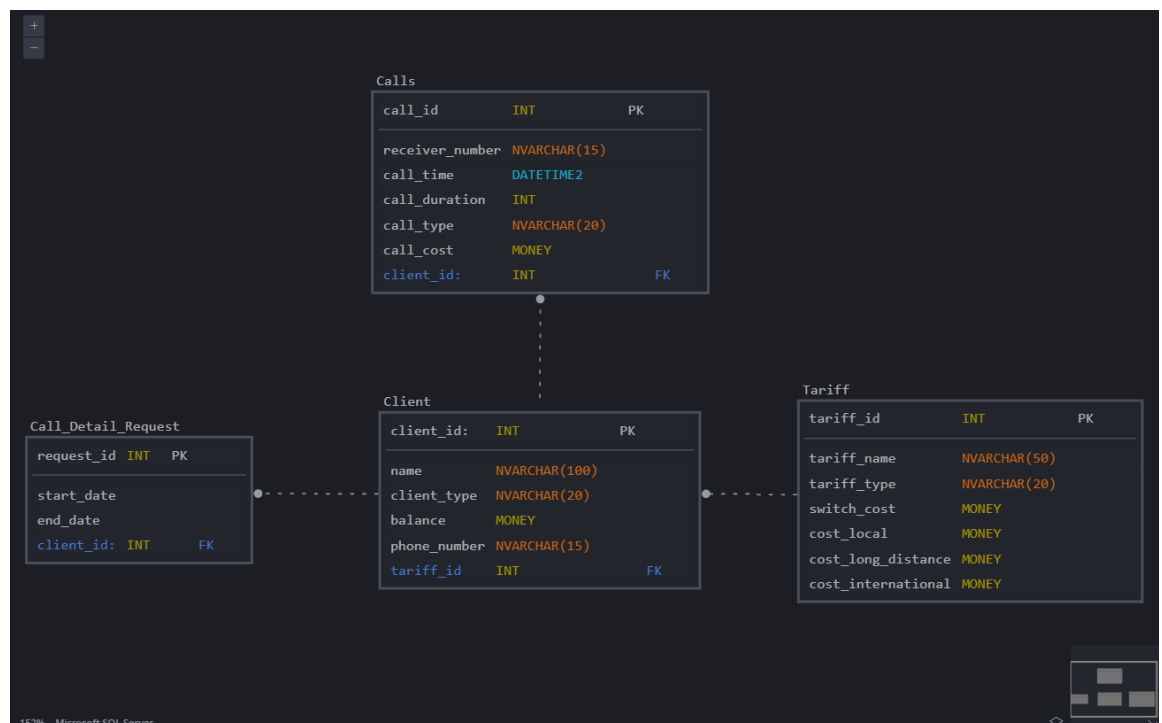


Рис.3. Физическая модель базы данных.

Создание БД в Microsoft SQL Server

Чтобы диаграмма была построена в Microsoft SQL Server, нужно создать БД и создать там таблицы, с необходимыми нам значениями. Будем использовать SQL-запросы.

Создаём БД с названием MTS (рис. 1).

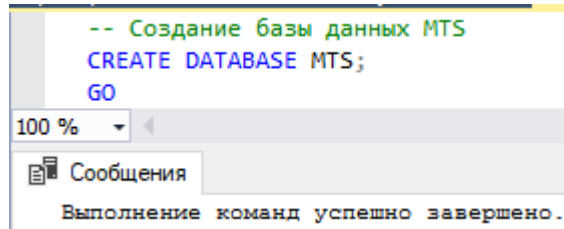


Рис. 1. Создание БД

Теперь перейдём к созданию необходимых нам таблиц с параметрами.

Таблица для Тарифов (рис. 2)

```
CREATE TABLE Tariff (
    tariff_id INT IDENTITY(1,1) PRIMARY KEY,          -- Идентификатор тарифа
    tariff_name NVARCHAR(50) NOT NULL,                 -- Название тарифа
    tariff_type NVARCHAR(20) CHECK(tariff_type IN ('Корпоративный', 'Некорпоративный')), -- Тип тарифа
    switch_cost MONEY NOT NULL,                        -- Стоимость перехода на тариф
    cost_local MONEY NOT NULL,                         -- Стоимость минуты по городу
    cost_long_distance MONEY NOT NULL,                 -- Стоимость минуты междугородний
    cost_international MONEY NOT NULL                 -- Стоимость минуты международный
);
GO
```

Рис. 2.

Таблица для Клиентов (рис. 3)

```
CREATE TABLE Client (
    client_id INT IDENTITY(1,1) PRIMARY KEY,          -- Идентификатор клиента
    name NVARCHAR(100) NOT NULL,                      -- Имя клиента
    client_type NVARCHAR(20) CHECK(client_type IN ('Физическое', 'Юридическое')), -- Тип клиента
    balance MONEY NOT NULL DEFAULT 0,                 -- Баланс клиента (по умолчанию 0)
    phone_number NVARCHAR(15) UNIQUE NOT NULL,        -- Номер телефона клиента
    tariff_id INT,                                     -- Идентификатор тарифа клиента
    FOREIGN KEY (tariff_id) REFERENCES Tariff(tariff_id) -- Связь с тарифом
    ON DELETE SET NULL -- Если тариф удалён, ставим NULL в поле тариф клиента
);
GO
```

Рис. 3.

Таблица для Звонков (рис. 4)

```
CREATE TABLE Calls (
    call_id INT IDENTITY(1,1) PRIMARY KEY,            -- Идентификатор звонка
    client_id INT,                                     -- Идентификатор клиента
    receiver_number NVARCHAR(15) NOT NULL,            -- Номер абонента, которому звонили
    call_time DATETIME2 NOT NULL,                     -- Время звонка (используем DATETIME2)
    call_duration INT NOT NULL,                      -- Длительность звонка (в минутах)
    call_type NVARCHAR(20) CHECK(call_type IN ('По городу', 'Междугородний', 'Международный')), -- Тип соединения
    call_cost MONEY NOT NULL DEFAULT 0,               -- Стоимость звонка (по умолчанию 0)
    FOREIGN KEY (client_id) REFERENCES Client(client_id) -- Связь с клиентом
    ON DELETE CASCADE -- Если клиент удалён, все его звонки тоже удаляются
);
GO
```

Рис. 4.

Таблица для Запросов детализации звонков (рис. 5)

```
-- Таблица для запросов детализации звонков
CREATE TABLE Call_Detail_Request (
    request_id INT IDENTITY(1,1) PRIMARY KEY, -- Идентификатор запроса
    client_id INT, -- Идентификатор клиента
    start_date DATE NOT NULL, -- Начальная дата для детализации
    end_date DATE NOT NULL, -- Конечная дата для детализации
    FOREIGN KEY (client_id) REFERENCES Client(client_id) -- Связь с клиентом
    ON DELETE CASCADE -- Если клиент удалён, запросы его детализации тоже удаляются
);
GO
```

Рис. 5.

Выполняем запрос, для БД MTS (рис. 6).

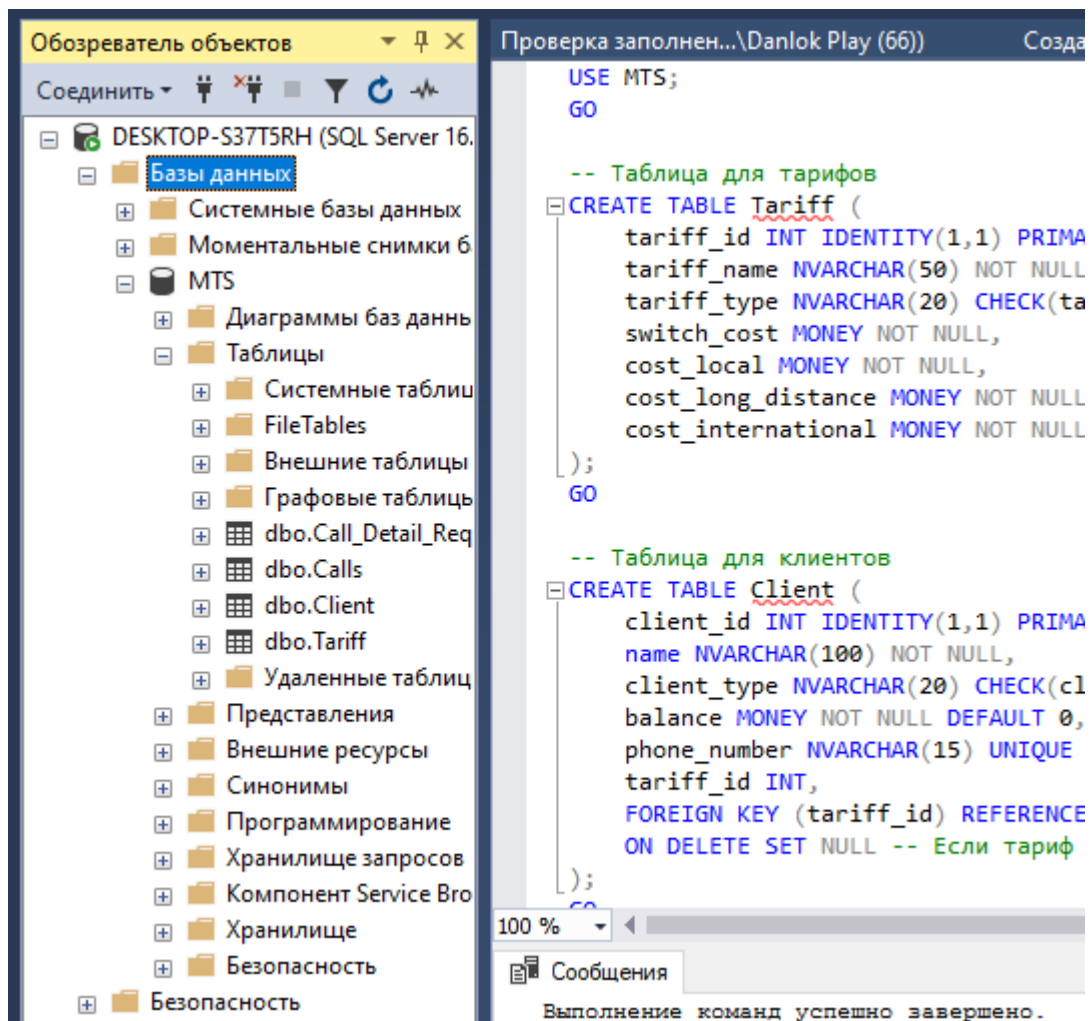


Рис. 6.

Как видим, в обозревателе решений появилась БД с нашими таблицами.

Теперь можем сделать диаграмму в Microsoft SQL Server, и посмотреть, что получилось (рис. 7).

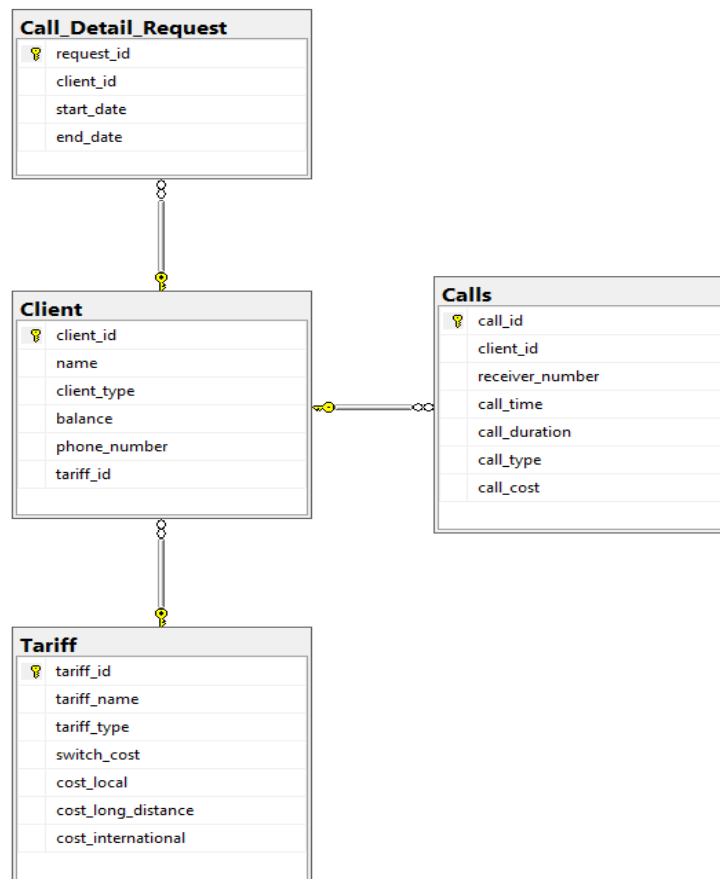


Рис. 7. Диаграмма в Microsoft SQL Server

Описание ограничений

При создании таблиц, мы использовали следующие ограничения:

1) Значения по умолчанию. DEFAULT

У нас будет начальный баланс у клиента 0, и по умолчанию стоимость совершённого звонка, также будет 0 (рис. 8).

```

balance MONEY NOT NULL DEFAULT 0,           -- Баланс клиента (по умолчанию 0)
call_cost MONEY NOT NULL DEFAULT 0,          -- Стоимость звонка (по умолчанию 0)
  
```

Рис. 8. Значения по умолчанию.

2) Ограничения на вводимые данные. CHECK

У нас будут ограничения на вводимые данные для Типа тарифа ('Корпоративный', 'Некорпоративный'), Типа клиента ('Физическое', 'Юридическое') и Типа звонка ('По городу', 'Междугородний', 'Международный') (рис. 8.)

```

tariff_type NVARCHAR(20) CHECK(tariff_type IN ('Корпоративный', 'Некорпоративный')), -- Тип тарифа (Проверка ограничения на вводимые данные)
client_type NVARCHAR(20) CHECK(client_type IN ('Физическое', 'Юридическое')), -- Тип клиента (Проверка ограничения на вводимые данные)
call_type NVARCHAR(20) CHECK(call_type IN ('По городу', 'Междугородний', 'Международный')), -- Тип соединения (Проверка ограничения на вводимые данные)
  
```

Рис. 8. Ограничения на вводимые данные

3) Правила удаления. DELETE

```
ON DELETE CASCADE -- Если клиент удалён, все его звонки тоже удаляются
ON DELETE CASCADE -- Если клиент удалён, запросы его детализации тоже удаляются
```

Рис. 9. ON DELETE CASCADE

Здесь в таблицах Calls и Call_Detail_Request используется ON DELETE CASCADE для связи с Client. Это значит: если удалить клиента из таблицы Client, то все его звонки (Calls) и запросы на детализацию (Call_Detail_Request) тоже автоматически удалятся.

```
ON DELETE SET NULL -- Если тариф удалён, ставим NULL в поле тариф клиента
```

Рис. 10. ON DELETE SET NULL

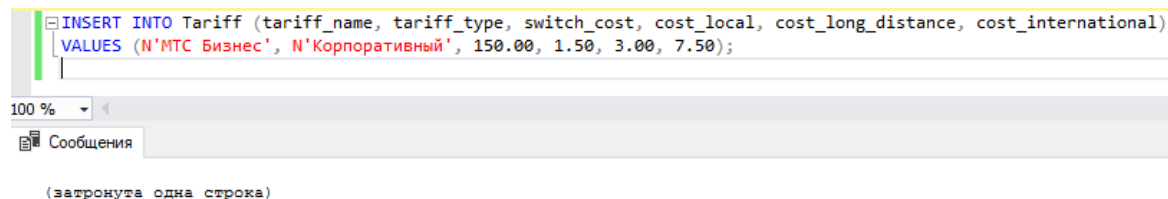
Здесь в таблице Client установлено правило ON DELETE SET NULL для tariff_id. Это значит: если удалить тариф из таблицы Tariff, у всех клиентов, которые были на этом тарифе, поле tariff_id станет NULL, но сами записи клиентов останутся в таблице.

Запросы на модификацию данных

Команда	Назначение
INSERT	Добавить новую запись
UPDATE	Изменить существующую запись
DELETE	Удалить запись
SELECT	Получить записи из таблицы

1) INSERT

Добавление тарифа (рис. 11)



```
INSERT INTO Tariff (tariff_name, tariff_type, switch_cost, cost_local, cost_long_distance, cost_international)
VALUES (N'МТС Бизнес', N'Корпоративный', 150.00, 1.50, 3.00, 7.50);
```

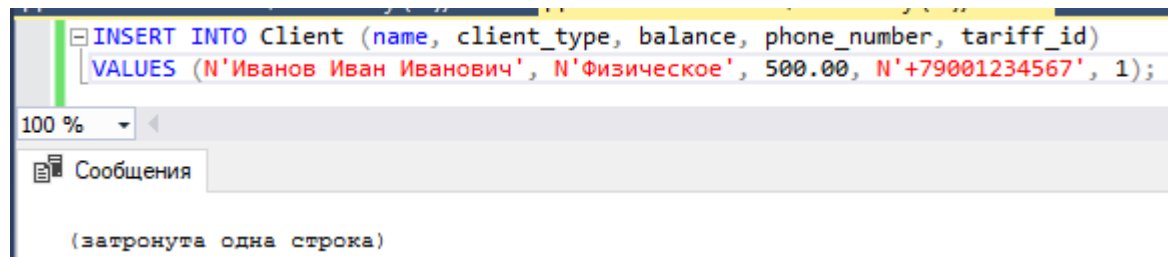
100 %

Сообщения

(затронута одна строка)

Рис. 11.

Добавление клиента (рис 12.)



```
INSERT INTO Client (name, client_type, balance, phone_number, tariff_id)
VALUES (N'Иванов Иван Иванович', N'Физическое', 500.00, N'+79001234567', 1);
```

100 %

Сообщения

(затронута одна строка)

Рис. 12.

Добавление звонка (рис. 13)

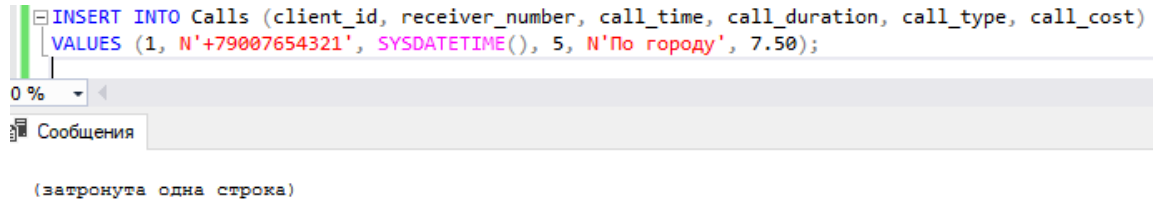


Рис. 13.

Посмотрим теперь содержание таблиц (рис. 15).

Для этого выполним следующий запрос (рис. 14)

```
USE MTS;
GO

-- Проверка данных в таблице Tariff
SELECT * FROM Tariff;
GO

-- Проверка данных в таблице Client
SELECT * FROM Client;
GO

-- Проверка данных в таблице Calls
SELECT * FROM Calls;
GO

-- Проверка данных в таблице Call_Detail_Request
SELECT * FROM Call_Detail_Request;
GO
```

Рис. 14. Просмотр данных в таблицах

Результат.

Результаты		Сообщения					
tariff_id	tariff_name	tariff_type	switch_cost	cost_local	cost_long_distance	cost_international	
1	МТС Бизнес	Корпоративный	150,00	1,50	3,00	7,50	

client_id	name	client_type	balance	phone_number	tariff_id
1	Иванов Иван Иванович	Физическое	500,00	+79001234567	1

call_id	client_id	receiver_number	call_time	call_duration	call_type	call_cost
1	1	+79007654321	2025-04-28 02:37:13.2403970	5	По городу	7,50

request_id	client_id	start_date	end_date
------------	-----------	------------	----------

Рис. 15. Данные в таблицах.

2) UPDATE

Обновить баланс клиента. Этот запрос увеличит баланс клиента с номером +79001234567 на 200 рублей.

```

UPDATE Client
SET balance = balance + 200
WHERE phone_number = N'+79001234567';

```

Рис. 16. Обновление баланса клиента.

Проверка. До:

	client_id	name	client_type	balance	phone_number	tariff_id
1	1	Иванов Иван Иванович	Физическое	500,00	+79001234567	1

После:

	client_id	name	client_type	balance	phone_number	tariff_id
1	1	Иванов Иван Иванович	Физическое	700,00	+79001234567	1

Изменить стоимость перехода на тариф. Этот запрос изменяет стоимость перехода (switch_cost) для тарифа с tariff_id = 1 на 100 рублей.

```

UPDATE Tariff
SET switch_cost = 100.00
WHERE tariff_id = 1;

```

Рис. 17. Изменение стоимости перехода на тариф.

Проверка. До:

	tariff_id	tariff_name	tariff_type	switch_cost	cost_local	cost_long_distance	cost_international
1	1	МТС Бизнес	Корпоративный	150,00	1,50	3,00	7,50

После:

	tariff_id	tariff_name	tariff_type	switch_cost	cost_local	cost_long_distance	cost_international
1	1	МТС Бизнес	Корпоративный	100,00	1,50	3,00	7,50

Изменить тариф у клиента. Этот запрос меняет тариф клиента **только если** тип клиента и тип тарифа совпадают:

Физическое лицо → Некорпоративный тариф

Юридическое лицо → Корпоративный тариф

Если не подходит — тариф не изменится.

```

UPDATE Client
SET tariff_id = 4
FROM Client c
JOIN Tariff t ON t.tariff_id = 4
WHERE c.client_id = 1
AND (
    (c.client_type = N'Физическое' AND t.tariff_type = N'Некорпоративный')
    OR
    (c.client_type = N'Юридическое' AND t.tariff_type = N'Корпоративный')
);

```

Рис. 18. Изменение тарифа у клиента.

Проверка. До:

	tariff_id	tariff_name	tariff_type	switch_cost	cost_local	cost_long_distance	cost_international
1	1	МТС Бизнес	Корпоративный	100,00	1,50	3,00	7,50
2	3	МТС Мой	Некорпоративный	150,00	1,50	3,00	7,50
3	4	МТС Один	Некорпоративный	150,00	1,50	3,00	7,50

	client_id	name	client_type	balance	phone_number	tariff_id
1	1	Иванов Иван Иванович	Физическое	700,00	+79001234567	3

После:

	tariff_id	tariff_name	tariff_type	switch_cost	cost_local	cost_long_distance	cost_international
1	1	МТС Бизнес	Корпоративный	100,00	1,50	3,00	7,50
2	3	МТС Мой	Некорпоративный	150,00	1,50	3,00	7,50
3	4	МТС Один	Некорпоративный	150,00	1,50	3,00	7,50

	client_id	name	client_type	balance	phone_number	tariff_id
1	1	Иванов Иван Иванович	Физическое	700,00	+79001234567	4

3) DELETE

Удаление всех звонков у клиента (рис. 19).

```
DELETE FROM Calls
WHERE client_id = 1;
```

Рис. 19. Удаление звонков у клиента.

Удаление клиента по номеру телефона (рис. 20).

```
DELETE FROM Client
WHERE phone_number = N'+79001234567';
```

Рис. 20. Удаление клиента по номеру телефона.

Удаление тарифа (рис. 21).

```
DELETE FROM Tariff
WHERE tariff_id = 3;
```

Рис. 21. Удаление тарифа.

Таблицы до выполнения запросов (рис. 22).

	tariff_id	tariff_name	tariff_type	switch_cost	cost_local	cost_long_distance	cost_international
1	1	МТС Бизнес	Корпоративный	100,00	1,50	3,00	7,50
2	3	МТС Мой	Некорпоративный	150,00	1,50	3,00	7,50
3	4	МТС Один	Некорпоративный	150,00	1,50	3,00	7,50

	client_id	name	client_type	balance	phone_number	tariff_id
1	1	Иванов Иван Иванович	Физическое	700,00	+79001234567	4

	call_id	client_id	receiver_number	call_time	call_duration	call_type	call_cost
1	1	1	+79007654321	2025-04-28 02:37:13.2403970	5	По городу	7,50

Рис. 22. Таблицы до выполнения запросов.

После выполнения этих запросов (рис. 23).

Результаты		Сообщения					
	tariff_id	tariff_name	tariff_type	switch_cost	cost_local	cost_long_distance	cost_intemational
1	1	МТС Бизнес	Корпоративный	100,00	1,50	3,00	7,50
2	4	МТС Один	Некорпоративный	150,00	1,50	3,00	7,50

client_id	name	client_type	balance	phone_number	tariff_id
-----------	------	-------------	---------	--------------	-----------

call_id	client_id	receiver_number	call_time	call_duration	call_type	call_cost
---------	-----------	-----------------	-----------	---------------	-----------	-----------

Рис. 23. Таблицы после выполнения команд.

Всё сработало корректно.

Запросы на выборку данных. 10 запросов SELECT

Перед выполнением этих запросов, заполним таблицы (рис. 24).

Результаты		Сообщения					
59	59	MTS-Стандарт	Некорпоративный	200,00	1,90	2,90	4,20
60	60	MTS-Про	Корпоративный	310,00	1,60	2,60	3,80
61	61	MTS-Гига-Про	Корпоративный	400,00	1,30	2,30	3,90
62	62	MTS-Люкс	Корпоративный	260,00	2,10	3,10	4,60

	client_id	name	client_type	balance	phone_number	tariff_id
99	99	АО "НейроСистемы"	Юридическ...	1598,65	8916092531	13
100	100	ООО "Системные Решения"	Юридическ...	1683,55	8916096075	10
101	101	ООО "ТехСистемы"	Юридическ...	1355,45	8916093237	34
102	102	ООО "ПрогрессГрупп"	Юридическ...	1690,95	8916025904	16
103	103	ИП "Новиков Александр Вале...	Юридическ...	4176,78	8916041639	10
104	104	АО "КосмоГрупп"	Юридическ...	4057,48	8916095324	2
105	105	АО "ТехноСтрим"	Юридическ...	2462,98	8916014542	46

	call_id	client_id	receiver_number	call_time	call_duration	call_type	call_cost
2329	2329	10	891612345265	2024-07-21 10:55:00.0000000	18	Междугородний	59,40
2330	2330	52	891612345237	2024-03-11 20:28:00.0000000	12	По городу	37,20
2331	2331	42	891612345201	2024-05-18 11:03:00.0000000	5	По городу	8,50
2332	2332	60	891612345749	2024-07-16 08:24:00.0000000	14	Междугородний	28,00
2333	2333	42	891612345616	2024-06-30 11:46:00.0000000	11	Междугородний	29,70

	request_id	client_id	start_date	end_date
633	633	88	2024-04-21	2025-02-02
634	634	17	2024-03-21	2025-01-25
635	635	52	2024-03-08	2025-01-18
636	636	96	2024-04-16	2025-02-09
637	637	66	2024-03-05	2025-01-30
638	638	83	2024-04-22	2025-01-17

Рис. 24. Заполненные таблицы.

1) WHERE

```
USE MTS;
GO
SELECT * FROM Client
WHERE balance > 1500;
```

100 %

Результаты Сообщения

	client_id	name	client_type	balance	phone_number	tariff_id
1	68	АО "Технопарк"	Юридическое	4268,02	8916092315	47
2	70	ООО "Мегасистемы"	Юридическое	4660,72	8916099690	22
3	71	ИП "Тимофеев Михаил Александрович"	Юридическое	3032,00	8916092477	30
4	72	ИП "Смирнов Владимир Иванович"	Юридическое	1976,33	8916046585	17
5	73	ООО "ИнтерТех"	Юридическое	4278,76	8916000887	22
6	74	АО "Глобус-Тех"	Юридическое	4719,27	8916039835	28
7	75	ИП "Ковалев Алексей Владимирович"	Юридическое	3753,44	8916003294	25

Рис. 25. WHERE

Запрос выводит список клиентов, у которых баланс превышает 1500. Условие WHERE используется для фильтрации данных, чтобы выбрать только те записи, которые соответствуют указанному критерию.

2) ORDER BY

```
USE MTS;
GO
SELECT * FROM Tariff
ORDER BY switch_cost ASC;
```

100 %

Результаты Сообщения

	tariff_id	tariff_name	tariff_type	switch_cost	cost_local	cost_long_distance	cost_international
1	26	MTS-Смарт	Некорпоративный	90,00	2,10	3,10	4,50
2	31	MTS-Экспресс	Некорпоративный	90,00	2,10	3,10	4,60
3	54	MTS-Смарт	Некорпоративный	90,00	2,10	3,10	4,50
4	37	MTS-Смарт	Некорпоративный	100,00	2,20	3,20	4,60
5	39	MTS-Лайт	Некорпоративный	100,00	2,00	3,00	4,50
6	47	MTS-Смарт	Корпоративный	100,00	2,20	3,20	4,60
7	22	MTS-Смарт	Корпоративный	100,00	2,20	3,20	4,60
8	6	MTS-Экспресс	Корпоративный	100,00	2,20	3,20	4,70
9	16	MTS-Лайт	Корпоративный	110,00	2,10	3,10	4,60
10	49	MTS-Комфорт	Некорпоративный	110,00	2,30	3,30	4,60
11	51	MTS-Лайт	Корпоративный	110,00	2,10	3,10	4,60

Рис. 26. ORDER BY

Запрос выводит все данные из таблицы Tariff, сортируя результаты по стоимости перехода на тариф (switch_cost) от наименьшей к наибольшей.

3) Функция агрегации

<pre>USE MTS; GO SELECT SUM(call_cost) AS total_revenue FROM Calls;</pre>	
100 %	
Результаты	Сообщения
total_revenue	
1	106620,80

Рис. 27. Функция агрегации

Запрос подсчитывает общую сумму всех звонков (их стоимости) в таблице Calls и выводит её как total_revenue (общий доход).

4) GROUP BY + HAVING

<pre>USE MTS; GO SELECT call_type, COUNT(*) AS call_count FROM Calls GROUP BY call_type HAVING COUNT(*) >= 750;</pre>	
100 %	
Результаты	Сообщения
call_type	call_count
1 Междугородний	1016
2 По городу	756

Рис. 28. GROUP BY + HAVING

Запрос выводит количество звонков для каждого типа звонка, но только для тех типов, где количество звонков больше или равно 750.

5) CAST

<pre>USE MTS; GO SELECT name, CAST(balance AS INT) AS balance_int FROM Client;</pre>	
100 %	
Результаты	Сообщения
name	balance_int
1 Иванов Иван Иванович	802
2 Петров Петр Петрович	560
3 Сидоров Сергей Сергеевич	866
4 Кузнецов Николай Николаевич	622
5 Васильев Владимир Викторович	910
6 Морозов Михаил Михайлович	505
7 Попов Алексей Александрович	507
8 Александров Андрей Андреевич	236
9 Михайлов Денис Денисович	505

Рис. 29. CAST

Запрос выводит имена клиентов и их баланс, при этом баланс преобразуется в целое число (отбрасываются дробные значения, если они есть).

6) CASE

```
USE MTS;
GO
SELECT call_id, call_cost,
CASE
WHEN call_cost < 20 THEN 'Дешевый'
WHEN call_cost BETWEEN 20 AND 45 THEN 'Средний'
ELSE 'Дорогой'
END AS call_category
FROM Calls;
```

	call_id	call_cost	call_category
1	1	78.20	Дорогой
2	2	11.20	Дешевый
3	3	64.00	Дорогой
4	4	96.00	Дорогой
5	5	15.60	Дешевый
6	6	10.80	Дешевый
7	7	13.60	Дешевый
8	8	59.40	Дорогой
9	9	37.80	Средний
10	10	16.90	Дешевый
11	11	34.50	Средний
12	12	9.60	Дешевый

Рис. 30. CASE

Запрос выводит идентификатор звонка, стоимость звонка и категорию звонка (Дешевый, Средний или Дорогой) в зависимости от стоимости звонка.

7) EXISTS

```
USE MTS;
GO
SELECT t.tariff_id, t.tariff_name
FROM Tariff t
WHERE EXISTS (
SELECT 1 FROM Client c WHERE c.tariff_id = t.tariff_id
);
```

	tariff_id	tariff_name
1	1	MTS-Бизнес
2	2	MTS-Про
3	4	MTS-Гига-Про
4	5	MTS-Смарт
5	7	MTS-Стандарт
6	8	MTS-Люкс
7	9	MTS-Оптима+
8	10	MTS-Мег
9	11	MTS-Гибрид
10	13	MTS-Базовый
11	14	MTS-Плюс
12	15	MTS-Супер
13	16	MTS-Лайт
14	17	MTS-Максим
15	18	MTS-Комфорт
16	19	MTS-Эконом
17	20	MTS-Гига

Рис. 31. EXISTS

Запрос выводит тарифы, которые используются хотя бы одним клиентом. Если у тарифа есть хотя бы один клиент, он будет включён в результат.

8) Подзапрос

```
USE MTS;
GO
SELECT * FROM Client
WHERE balance > (SELECT AVG(balance) FROM Client);
```

	client_id	name	client_type	balance	phone_number	tariff_id
1	68	АО "Технопарк"	Юридическое	4268,02	8916092315	47
2	70	ООО "Мегасистемы"	Юридическое	4660,72	8916099690	22
3	71	ИП "Тимофеев Михаил Александрович"	Юридическое	3032,00	8916092477	30
4	72	ИП "Смирнов Владимир Иванович"	Юридическое	1976,33	8916046585	17
5	73	ООО "ИнтерТех"	Юридическое	4278,76	8916000887	22
6	74	АО "Глобус-Тех"	Юридическое	4719,27	8916039835	28
7	75	ИП "Ковалев Алексей Владимирович"	Юридическое	3753,44	8916003294	25
8	76	ООО "ПроектСистемы"	Юридическое	4533,09	8916000912	4
9	78	ООО "ЭкспертГрупп"	Юридическое	4551,14	8916079899	35
10	79	АО "ПроГрупп"	Юридическое	2091,45	8916069016	4
11	80	ООО "ВекторТех"	Юридическое	3229,08	8916088315	10
12	81	ООО "Вектор-Проект"	Юридическое	4429,93	8916067529	1

Рис. 32. Подзапрос

Запрос выводит всех клиентов, чьи балансы больше, чем средний баланс всех клиентов в базе данных.

9) JOIN

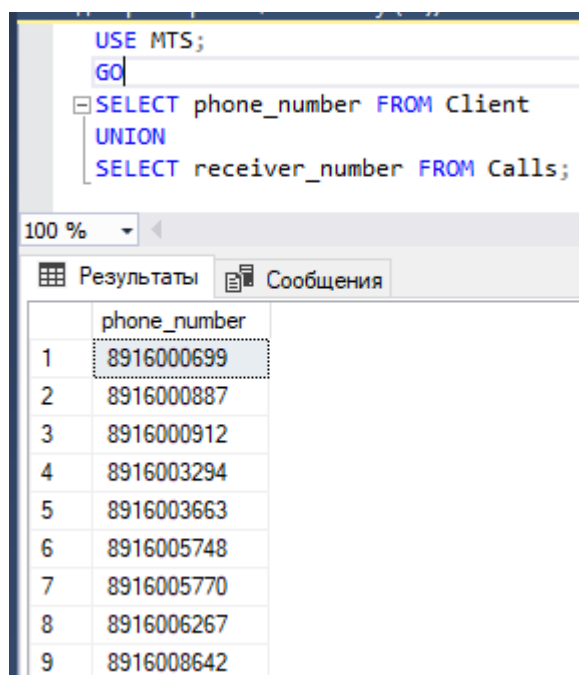
```
USE MTS;
GO
SELECT c.name, c.phone_number, t.tariff_name
FROM Client c
JOIN Tariff t ON c.tariff_id = t.tariff_id;
```

	name	phone_number	tariff_name
1	Иванов Иван Иванович	8916030626	MTS-Тарифище
2	Петров Петр Петрович	8916063036	MTS-Комфорт
3	Сидоров Сергей Сергеевич	8916083670	MTS-Максимум
4	Кузнецов Николай Николаевич	8916059159	MTS-Смарт
5	Васильев Владимир Викторович	8916072786	MTS-Супер
6	Морозов Михаил Михайлович	8916029936	MTS-Базовый
7	Попов Алексей Александрович	8916078791	MTS-Максимум
8	Александров Андрей Андреевич	8916010379	MTS-Ультра
9	Михайлов Денис Денисович	8916051762	MTS-Супер
10	Ковалев Максим Владимирович	8916045636	MTS-Гибрид
11	Григорьев Олег Олегович	8916033605	MTS-Тарифище
12	Чистяков Алексей Сергеевич	8916049437	MTS-Гига
13	Федоров Алексей Викторович	8916082150	MTS-Базовый
14	Гусев Алексей Валерьевич	8916041417	MTS-Смарт-Плюс
15	Жуков Сергей Павлович	8916031765	MTS-Базовый

Рис. 33. JOIN

Запрос выводит имена клиентов, их номера телефонов и название их тарифов. Для этого объединяются данные из таблиц Client и Tariff по полю tariff_id.

10) UNION



The screenshot shows a SQL query editor with the following query:

```
USE MTS;  
GO  
SELECT phone_number FROM Client  
UNION  
SELECT receiver_number FROM Calls;
```

Below the query, the results are displayed in a table with the column 'phone_number'.

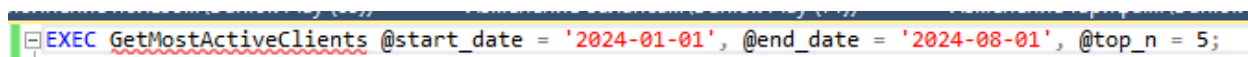
	phone_number
1	8916000699
2	8916000887
3	8916000912
4	8916003294
5	8916003663
6	8916005748
7	8916005770
8	8916006267
9	8916008642

Рис. 34. UNION

Запрос выводит уникальные номера телефонов, которые присутствуют либо среди клиентов (phone_number из Client), либо среди получателей звонков (receiver_number из Calls).

Хранимые процедуры

1) Хранимая процедура GetMostActiveClients позволяет вывести информацию о самых активных клиентах за определённый период. Для этого нужно передать дату начала и окончания периода, а также количество клиентов, которых нужно вывести в результатах. Процедура возвращает топ клиентов, совершивших наибольшее количество звонков, и сортирует их по убыванию активности.



```
EXEC GetMostActiveClients @start_date = '2024-01-01', @end_date = '2024-08-01', @top_n = 5;
```

Рис. 35. Запрос к процедуре 1.

Этот запрос вернёт топ-5 самых активных клиентов, которые совершили наибольшее количество звонков в период с 1 января 2024 года по 1 августа 2024 года.

```

USE MTS;
GO

IF OBJECT_ID('GetMostActiveClients', 'P') IS NOT NULL
    DROP PROCEDURE GetMostActiveClients;
GO

CREATE PROCEDURE GetMostActiveClients
    @start_date DATETIME2,
    @end_date DATETIME2,
    @top_n INT
AS
BEGIN
    SET NOCOUNT ON;

    SELECT TOP(@top_n)
        c.client_id,
        c.name,
        c.phone_number,
        COUNT(cl.call_id) AS total_calls
    FROM Client c
    LEFT JOIN Calls cl ON c.client_id = cl.client_id
    WHERE cl.call_time BETWEEN @start_date AND @end_date
    GROUP BY c.client_id, c.name, c.phone_number
    ORDER BY total_calls DESC;
END;
GO

```

Рис. 36. Процедура 1.

Результат:

EXEC GetMostActiveClients @start_date = '2024-01-01', @end_date = '2024-08-01', @top_n = 5;

	client_id	name	phone_number	total_calls
1	16	Крылов Игорь Николаевич	8916011756	33
2	29	Новиков Дмитрий Михайлович	8916015485	31
3	33	Смирнов Борис Евгеньевич	8916006267	30
4	34	Тимошенко Михаил Викторович	8916008642	30
5	94	ООО "НейроГрупп"	8916086174	29

2) Процедура `GetClientCallCostAnalysis` анализирует стоимость звонков клиентов за определённый период, который задаётся двумя параметрами — начальной и конечной датой. Она собирает информацию о стоимости звонков каждого клиента, разделяя их на городские, междугородные и международные. Для каждого клиента рассчитывается общая стоимость всех его звонков в указанный период. Результаты выводятся в виде списка с именем клиента, его номером телефона и детализацией стоимости звонков по категориям, с указанием общей стоимости. Данные сортируются по общей стоимости звонков в порядке убывания, то есть сначала идут клиенты с наибольшими расходами.

```

USE MTS;
GO

IF OBJECT_ID('GetClientCallCostAnalysis', 'P') IS NOT NULL
    DROP PROCEDURE GetClientCallCostAnalysis;
GO

CREATE PROCEDURE GetClientCallCostAnalysis
    @start_date DATE,          -- Начальная дата для анализа
    @end_date DATE             -- Конечная дата для анализа
AS
BEGIN
    SET NOCOUNT ON;

    -- Основной запрос для получения информации о стоимости звонков по клиентам
    SELECT
        c.client_id,
        c.name AS client_name,
        c.phone_number,
        SUM(CASE
            WHEN ca.call_type = 'По городу' THEN ca.call_cost
            ELSE 0
        END) AS local_calls_cost, -- Стоимость городских звонков
        SUM(CASE
            WHEN ca.call_type = 'Междугородний' THEN ca.call_cost
            ELSE 0
        END) AS long_distance_calls_cost, -- Стоимость междугородних звонков
        SUM(CASE
            WHEN ca.call_type = 'Международный' THEN ca.call_cost
            ELSE 0
        END) AS international_calls_cost, -- Стоимость международных звонков
        SUM(ca.call_cost) AS total_calls_cost -- Общая стоимость всех звонков
    FROM Client c
    JOIN Calls ca ON c.client_id = ca.client_id
    WHERE ca.call_time BETWEEN @start_date AND @end_date
    GROUP BY c.client_id, c.name, c.phone_number
    ORDER BY total_calls_cost DESC;
END;
GO

```

Рис. 37. Процедура 2.

EXEC GetClientCallCostAnalysis @start_date = '2024-01-01', @end_date = '2024-10-30';

	client_id	client_name	phone_number	local_calls_cost	long_distance_calls_cost	international_calls_cost	total_calls_cost
1	52	Мельников Сергей Александрович	8916019556	545,00	799,30	476,00	1820,30
2	16	Крылов Игорь Николаевич	8916011756	472,10	702,90	508,90	1683,90
3	70	ООО "Мегасистемы"	8916099690	546,80	586,40	464,20	1597,40
4	31	Павлов Валерий Викторович	8916039575	270,50	661,40	637,70	1569,60
5	29	Новиков Дмитрий Михайлович	8916015485	435,10	784,30	344,50	1563,90
6	46	Егорова Светлана Петровна	8916029286	560,60	520,60	430,00	1511,20
7	94	ООО "НейроГрупп"	8916086174	545,30	281,00	669,70	1496,00
8	54	Тимофеев Павел Анатольевич	8916005748	404,60	419,00	669,70	1493,30
9	57	Петрова Ирина Викторовна	8916018057	482,90	691,70	271,50	1446,10
10	65	Степанов Сергей Анатольевич	8916011228	707,60	582,30	152,80	1442,70
11	33	Смирнов Борис Евгеньевич	8916006267	280,20	884,80	271,70	1436,70

Рис. 38. Результат запроса к процедуре 2.

Триггеры