

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ  
БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ**

**СЕРГИЕНКО ЛЕВ ЭДУАРДОВИЧ**

**Дополнение к Лабораторной работе 6 по  
индивидуальному заданию студентов**

Отчет по Дополнение к Лабораторной работе 6 по  
индивидуальному заданию студентов

**Преподаватель**  
**Мороз А.С.**

**2024**

```

1  -- Задание 23: Создание 2 хранимых процедур с входными параметрами
2
3  -- 1. Получение информации о клиенте по его ID
4  -- Business value: быстро получать подробную информацию о клиенте
5
6  DROP PROC IF EXISTS GetClientByID;
7  GO
8  CREATE PROC GetClientByID
9  ... @ClientID NVARCHAR(255)
10 AS
11 BEGIN
12 ... SELECT
13 ... id AS [Идентификатор клиента],
14 ... name AS [Имя],
15 ... phone AS [Телефон],
16 ... email AS [Электронная почта],
17 ... discount AS [Скидка],
18 ... registration_date AS [Дата регистрации]
19 ... FROM
20 ... clients
21 ... WHERE
22 ... id = @ClientID;
23 END
24 GO
25
26 EXEC GetClientByID @ClientID = 'JUR001';
27 GO
28

```

**Results**    Messages

	Идентификатор клиента ▾	Имя ▾	Телефон ▾	Электронная почта ▾	Скидка ▾	Дата регистрации ▾
1	JUR001	Romashka LLC	+79001234567	contact@romashka.ru	5.00	2024-01-15

```
-- 2. Получение списка бронирований за указанный период
-- Business value: отслеживать загрузку гостиницы и планировать ресурсы.

DROP PROC IF EXISTS GetBookingsByDateRange;
GO
CREATE PROC GetBookingsByDateRange
    @StartDate DATE,
    @EndDate DATE
AS
BEGIN
    SELECT
        b.id AS [ID бронирования],
        c.name AS [Клиент],
        r.id AS [Номер комнаты],
        b.start_date AS [Дата начала],
        b.end_date AS [Дата окончания],
        bs.name AS [Статус бронирования]
    FROM
        bookings b
    JOIN clients c ON b.booker_id = c.id
    JOIN rooms r ON b.room_id = r.id
    JOIN booking_statuses bs ON b.status_id = bs.id
    WHERE
        b.start_date >= @StartDate
        AND b.end_date <= @EndDate;
END
GO
EXEC GetBookingsByDateRange @StartDate = '2024-12-01', @EndDate = '2024-12-31';
GO
```

ults Messages

ID бронирования	Клиент	Номер комнаты	Дата начала	Дата окончания	Статус бронирования
16	Sidorov SP	16	2024-12-01	2024-12-06	Confirmed
17	Roza LLC	17	2024-12-03	2024-12-08	Pending
18	Gerbera JSC	18	2024-12-05	2024-12-10	Cancelled
19	Ivanov SP	19	2024-12-07	2024-12-12	Confirmed
20	Chrysanthemum LLC	20	2024-12-09	2024-12-14	Completed
29	Ekaterina Zakharova	29	2024-12-02	2024-12-05	Completed
30	Sergey Vorobyov	30	2024-12-06	2024-12-09	Confirmed

-- Задание 24: Создание 2 хранимых процедур с выходными параметрами

-- 1. Получение общего количества бронирований для клиента

-- Business value: оценивать активность клиентов

```
DROP PROC IF EXISTS GetTotalBookingsByClient;
GO
CREATE PROC GetTotalBookingsByClient
    @ClientID NVARCHAR(255),
    @TotalBookings INT OUTPUT
AS
BEGIN
    SELECT
        @TotalBookings = COUNT(*)
    FROM
        bookings
    WHERE
        booker_id = @ClientID;
END
GO

DECLARE @Total INT;
EXEC GetTotalBookingsByClient @ClientID = 'JUR001', @TotalBookings = @Total OUTPUT;
SELECT @Total AS [Общее количество бронирований];
GO
```

ults Messages

Общее количество бронирований

2

```

-- 2. Получение средней цены номера в здании
-- Business value: Помогает в анализе
DROP PROC IF EXISTS GetAverageRoomPriceByBuilding;
GO
CREATE PROC GetAverageRoomPriceByBuilding
    @BuildingID INT,
    @AveragePrice DECIMAL(10,2) OUTPUT
AS
BEGIN
    SELECT
        @AveragePrice = AVG(price)
    FROM
        rooms
    WHERE
        building_id = @BuildingID;
END
GO

DECLARE @AvgPrice DECIMAL(10,2);
EXEC GetAverageRoomPriceByBuilding @BuildingID = 1, @AveragePrice = @AvgPrice OUTPUT;
SELECT @AvgPrice AS [Средняя цена номера];
GO

```

## 5 Messages

средняя цена номера ▾

2864.24

-- Задание 25: Создание 2 определяемых пользователем скалярных функций

-- 1. Вычисление продолжительности бронирования в днях

-- Business value: оценить длительность пребывания клиентов для аналитики и планирования

```
IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'dbo.CalculateBookingDuration') AND type = 'FUNCTION')
DROP FUNCTION dbo.CalculateBookingDuration;
GO
CREATE FUNCTION dbo.CalculateBookingDuration
(
    @StartDate DATE,
    @EndDate DATE
)
RETURNS INT
AS
BEGIN
    RETURN DATEDIFF(DAY, @StartDate, @EndDate);
END
GO
SELECT
    id AS [ID бронирования],
    dbo.CalculateBookingDuration(start_date, end_date) AS [Продолжительность (дни)]
FROM
    bookings;
GO
```

ts Messages

ID бронирования	Продолжительность (дни)
1	4
2	4
3	5
4	5
5	5
6	5
7	5
8	5
9	5
10	5
11	5
12	5
13	-

-- 2. Определение VIP-клиента на основе скидки  
 -- Business value: автоматически классифицировать клиентов для предоставления специальных предложений.

```

DROP FUNCTION IF EXISTS dbo.IsVIPClient;
GO
CREATE FUNCTION dbo.IsVIPClient
(
  @Discount DECIMAL(5,2)
)
RETURNS BIT
AS
BEGIN
  RETURN CASE
    WHEN @Discount >= 10.00 THEN 1
    ELSE 0
  END;
END
GO

SELECT
  id AS [ID клиента],
  name AS [Имя],
  discount AS [Скидка],
  dbo.IsVIPClient(discount) AS [VIP]
FROM
  clients;
GO

```

#### Messages

ID клиента	Имя	Скидка	VIP
AB1234567	Ivan Ivanov	0.00	0
CD7654321	Petr Petrov	0.00	0
client123	John Doe	5.00	0
EF2345678	Anna Smirnova	0.00	0
GH8765432	Olga Sidorova	0.00	0
IJ3456789	Maria Kuznetsova	0.00	0
JUR001	Romashka LLC	5.00	0
JUR002	Vasilek JSC	10.00	1
JUR003	Petrov SP	0.00	0
JUR004	Liliya LLC	7.50	0
JUR005	Tulpan JSC	12.00	1
JUR006	Sidorov SP	0.00	0
JUR007	Peta LLC	0.00	0

```

1  -- 1. Список активных бронирований
2  -- Business value: быстро получать актуальные бронирования для управления загрузкой гостиницы.
3
4  DROP FUNCTION IF EXISTS dbo.GetActiveBookings;
5  GO
6  CREATE FUNCTION dbo.GetActiveBookings()
7  RETURNS TABLE
8  AS
9  RETURN
10 (
11     SELECT
12         b.id AS [ID бронирования],
13         c.name AS [Клиент],
14         r.id AS [Номер комнаты],
15         b.start_date AS [Дата начала],
16         b.end_date AS [Дата окончания],
17         bs.name AS [Статус]
18     FROM
19         bookings b
20         JOIN clients c ON b.booker_id = c.id
21         JOIN rooms r ON b.room_id = r.id
22         JOIN booking_statuses bs ON b.status_id = bs.id
23     WHERE
24         GETDATE() BETWEEN b.start_date AND b.end_date
25 );
26 GO

```

**sults** Messages

ID бронирования	Клиент	Номер комнаты	Дата начала	Дата окончания	Статус
19	Ivanov SP	19	2024-12-07	2024-12-12	Confirmed
20	Chrysanthemum LLC	20	2024-12-09	2024-12-14	Completed



```

-- 2. Список услуг, используемых в конкретном бронировании
-- Business value: Обеспечивает детальный отчет о дополнительных услугах, использованных клиентами.
DROP FUNCTION IF EXISTS dbo.GetServicesByBooking;
GO
CREATE FUNCTION dbo.GetServicesByBooking
(
    @BookingID INT
)
RETURNS TABLE
AS
RETURN
(
    SELECT
        s.name AS [Услуга],
        e.amount AS [Сумма],
        e.expense_date AS [Дата расхода]
    FROM
        expenses e
    JOIN services s ON e.service_id = s.id
    WHERE
        e.booking_id = @BookingID
);
GO
SELECT * FROM dbo.GetServicesByBooking(1);
GO

```

Results Messages

Услуга	Сумма	Дата расхода
Cinema Session	500.00	2024-11-26
Breakfast	300.00	2024-11-26
Gym	400.00	2024-11-26