

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ
БЕЛАРУСЬ**

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

СЕРГИЕНКО ЛЕВ ЭДУАРДОВИЧ

**Дополнение к Лабораторной работе 7 по
индивидуальному заданию студентов**

Отчет по Дополнение к Лабораторной работе 7 по
индивидуальному заданию студентов

Преподаватель
Мороз А.С.

2024

```

4  -- AFTER INSERT ТРИГГЕРЫ
5
6  -- 27.1. AFTER INSERT триггер на таблицу clients
7  -- При добавлении клиента-физлица присваивать ему приветственную скидку 5% если discount еще 0.00.
8  DROP TRIGGER IF EXISTS trg_AfterInsert_Clients;
9  GO
10 CREATE TRIGGER trg_AfterInsert_Clients
11 ON clients
12 AFTER INSERT
13 AS
14 BEGIN
15     SET NOCOUNT ON;
16
17     UPDATE c
18     SET c.discount = 5.00
19     FROM clients c
20     JOIN inserted i ON i.id = c.id
21     JOIN clients_types ct ON ct.id = i.type_id
22     WHERE ct.name = 'Individual' AND c.discount = 0.00;
23 END;
24 GO
25
26 INSERT INTO clients (id, type_id, name, phone, email, discount, registration_date) VALUES
27 ('TEST1', 1, 'Romashka LLC', '+79001234567', 'contact@romashka.ru', 0.00, '2024-01-15');
28
29 SELECT id, discount FROM clients WHERE id = 'TEST1';
30
31 DELETE FROM clients WHERE id = 'TEST1';
32
33 -- 27.2 AFTER UPDATE

```

Results Messages

id	discount
TEST1	5.00

```

-- 27.2 AFTER UPDATE
-- Описание запроса: При обновлении статуса бронирования на 'Completed' (status_id=4), увеличивает рейтинг клиента.
-- Business value: Вознаграждает клиентов за завершённые бронирования, поощряя их к дальнейшим действиям.
DROP TRIGGER IF EXISTS trg_AfterUpdate_Bookings;
GO
CREATE TRIGGER trg_AfterUpdate_Bookings
ON bookings
AFTER UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    UPDATE c
    SET c.discount = CASE
        WHEN c.discount + 1 <= 20 THEN c.discount + 1
        ELSE 20
    END
    FROM clients c
    JOIN inserted i ON i.booker_id = c.id
    JOIN deleted d ON d.id = i.id
    WHERE i.status_id = 4 AND d.status_id <> 4;
END;
GO

UPDATE bookings
SET status_id = 4
WHERE id = 1;
GO

SELECT id, discount FROM clients WHERE id = 'JUR001';

```

Results Messages

id	discount
JUR001	7.00

```

-- 27.3 AFTER DELETE
-- Описание запроса: При удалении клиента, записывает информацию о клиенте в таблицу archived_clients.
-- Business value: Позволяет сохранять информацию о удаленных клиентах для анализа
DROP TABLE IF EXISTS archived_clients;
GO
CREATE TABLE archived_clients (
    id NVARCHAR(255) PRIMARY KEY,
    type_id INT,
    name NVARCHAR(255) NOT NULL,
    phone NVARCHAR(255) NOT NULL,
    email NVARCHAR(255) NOT NULL,
    discount DECIMAL(5,2) DEFAULT 0.0 NOT NULL,
    registration_date DATE DEFAULT GETDATE() NOT NULL,
);
GO
DROP TRIGGER IF EXISTS trg_AfterDelete_Clients;
GO
CREATE TRIGGER trg_AfterDelete_Clients
ON clients
AFTER DELETE
AS
BEGIN
    SET NOCOUNT ON;
    INSERT INTO archived_clients (id, type_id, name, phone, email, discount, registration_date)
    SELECT id, type_id, name, phone, email, discount, registration_date
    FROM deleted;
END;
GO
INSERT INTO clients (id, type_id, name, phone, email, discount, registration_date) VALUES
('TEST1', 1, 'Romashka LLC', '+79001234567', 'contact@romashka.ru', 0.00, '2024-01-15');
DELETE FROM clients
WHERE id = 'TEST1';
GO
SELECT * FROM archived_clients;
GO

```

ults Messages

id	type_id	name	phone	email	discount	registration_date
TEST1	1	Romashka LLC	+79001234567	contact@romashka.ru	5.00	2024-01-15

```
-- INSTEAD OF INSERT

-- Задание
-- 28.1 INSTEAD OF INSERT
-- Описание запроса: При вставке нового номера, если цена меньше 500, устанавливает минимальную цену 500.
-- Business value: Обеспечивает соблюдение минимальных цен для номеров.

DROP TRIGGER IF EXISTS trg_InsteadOfInsert_Rooms;
GO

CREATE TRIGGER trg_InsteadOfInsert_Rooms
ON rooms
INSTEAD OF INSERT
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO rooms (building_id, bed_num, floor, price)
    SELECT building_id, bed_num, floor,
           CASE WHEN price < 500 THEN 500 ELSE price END
    FROM inserted;
END;
GO

INSERT INTO rooms (building_id, bed_num, floor, price)
VALUES (1, 5, 2, 400.00);
GO

SELECT TOP 1 * FROM rooms ORDER BY id DESC;
GO

-- 28.2 INSTEAD OF UPDATE триггер на представление vw_BookingClient
```

Results Messages

id	building_id	bed_num	floor	price
54	1	5	2	500.00

```
-- 28.2 INSTEAD OF UPDATE триггер на представление vw_BookingClient
-- Описание запроса: При обновлении дат бронирования через представление проверяет, что start_date <= end_date.
-- Business value: Обеспечивает корректность данных о периодах бронирований.
```

```
DROP TRIGGER IF EXISTS trg_InsteadOfUpdate_vw_BookingClient;
GO

CREATE TRIGGER trg_InsteadOfUpdate_vw_BookingClient
ON vw_BookingClient
INSTEAD OF UPDATE
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM inserted
        WHERE start_date > end_date
    )
    BEGIN
        RAISERROR('start_date cannot be after end_date.', 16, 1);
        ROLLBACK TRANSACTION;
        RETURN;
    END

    UPDATE b
    SET b.start_date = i.start_date,
        b.end_date = i.end_date,
        b.status_id = i.status_id
    FROM bookings b
    JOIN inserted i ON b.id = i.booking_id;
```

Results Messages

id	room_id	booker_id	start_date	end_date	status_id	booking_date
1	1	JUR001	2024-12-21	2024-12-26	4	2024-11-26

```
-- 28.3 INSTEAD OF DELETE
-- Описание запроса: Разрешает удаление расходов только если они были сделаны более 30 дней назад.
-- Business value: Предотвращает удаление актуальных расходов
```

```
DROP TRIGGER IF EXISTS trg_InsteadOfDelete_Expenses;
GO
CREATE TRIGGER trg_InsteadOfDelete_Expenses
ON expenses
INSTEAD OF DELETE
AS
BEGIN
    SET NOCOUNT ON;

    IF EXISTS (
        SELECT 1
        FROM deleted
        WHERE DATEDIFF(DAY, expense_date, GETDATE()) <= 30
    )
    BEGIN
        RAISERROR('Cannot delete expenses made within the last 30 days.', 16, 1);
        ROLLBACK TRANSACTION;
        RETURN;
    END

    DELETE e
    FROM expenses e
    JOIN deleted d ON e.id = d.id;
END;
GO

BEGIN TRY
    DELETE FROM expenses
    WHERE id = 1;
END TRY
BEGIN CATCH
    PRINT ERROR_MESSAGE();
END CATCH
GO
```

sages

```
2:25:42 PM    Started executing query at Line 203
               Commands completed successfully.
2:25:42 PM    Started executing query at Line 206
               Commands completed successfully.
2:25:42 PM    Started executing query at Line 229
               (0 rows affected)
               Cannot delete expenses made within the last 30 days.
               Total execution time: 00:00:00.016
```

```
-- ТРАНЗАКЦИИ

-- 29.1 Транзакция создания нового бронирования и обновления скидки клиента
-- Описание запроса: Создает новое бронирование и одновременно обновляет скидку клиента в рамках одной транзакции.
-- Business value: Обеспечивает согласованность данных при выполнении связанных операций
BEGIN TRAN AddBookingAndUpdateDiscount

DECLARE @client_id NVARCHAR(255) = 'AB1234567';
DECLARE @room_id INT = 2;

INSERT INTO bookings (room_id, booker_id, start_date, end_date, status_id)
VALUES (@room_id, @client_id, '2024-12-20', '2024-12-25', 1);

UPDATE clients
SET discount = CASE
    WHEN discount < 10 THEN discount + 1
    ELSE discount
END
WHERE id = @client_id;

COMMIT TRAN AddBookingAndUpdateDiscount;
GO

SELECT * FROM bookings WHERE booker_id = 'AB1234567' ORDER BY id DESC;
GO

SELECT id, name, discount FROM clients WHERE id = 'AB1234567';
GO

-- 29.2 Транзакция добавления нового клиента и бронирования для него
```

ts Messages

id	room_id	booker_id	start_date	end_date	status_id	booking_date
66	2	AB1234567	2024-12-20	2024-12-25	1	2024-12-12
36	2	AB1234567	2024-12-20	2024-12-25	1	2024-12-12
35	2	AB1234567	2024-12-20	2024-12-25	1	2024-12-12
34	2	AB1234567	2024-12-20	2024-12-25	1	2024-12-11
21	21	AB1234567	2024-11-01	2024-11-03	2	2024-11-26

id	name	discount
AB1234567	Тимофей	4.00

```

8
9 -- 29.2 Транзакция добавления нового клиента и бронирования для него
0 -- Описание запроса: Добавляет нового клиента и создает для него бронирование в рамках одной транзакции
1 BEGIN TRAN AddClientAndBooking
2
3 DECLARE @newClientId NVARCHAR(255) = 'NEW123';
4 DECLARE @room_id INT = 3;
5
6 INSERT INTO clients (id, type_id, name, phone, email)
7 VALUES (@newClientId, 1, 'New Client', '+79991234567', 'newclient@example.com');
8
9 INSERT INTO bookings (room_id, booker_id, start_date, end_date, status_id)
0 VALUES (@room_id, @newClientId, '2024-12-26', '2024-12-28', 1);
1
2 COMMIT TRAN AddClientAndBooking;
3 GO
4
5 SELECT * FROM clients WHERE id = 'NEW123';
6 GO
7
8 SELECT * FROM bookings WHERE booker_id = 'NEW123';
9 GO
0
1

```

Results Messages

id	type_id	name	phone	email	discount	registration_date
NEW123	1	New Client	+79991234567	newclient@example.com	5.00	2024-12-12

id	room_id	booker_id	start_date	end_date	status_id	booking_date
67	3	NEW123	2024-12-26	2024-12-28	1	2024-12-12

```

-- 30.1 Транзакция вставки услуги с несуществующим типом
-- Описание запроса: Пытается вставить услугу с несуществующим type_id, обрабатывает ошибку внешнего ключа.
BEGIN TRAN InsertServiceWithInvalidType
BEGIN TRY
... INSERT INTO services (type_id, name, price) VALUES (9999, 'Invalid Service', 500.00);
... COMMIT TRAN InsertServiceWithInvalidType;
END TRY
BEGIN CATCH
... ROLLBACK TRAN InsertServiceWithInvalidType;
... PRINT 'Error in InsertServiceWithInvalidType transaction: ' + ERROR_MESSAGE();
END CATCH
GO

```

Messages

```

.:27:15 PM Started executing query at Line 295
(0 rows affected)
Error in InsertServiceWithInvalidType transaction: The INSERT statement conflicted with the FOREIGN KEY const
B", table "dbo.service_types", column 'id'.
Total execution time: 00:00:00.002

```

```
306 -- 30.2 Транзакция добавления бронирования с end_date < start_date
307 -- Описание запроса: Пытается добавить бронирование, где end_date раньше start_date, проверяет логические ограничения.
308 BEGIN TRAN InsertInvalidBookingDates
309 BEGIN TRY
310     DECLARE @room_id INT = 5;
311     DECLARE @client_id NVARCHAR(255) = 'EF2345678';
312     DECLARE @start_date DATE = '2024-12-10';
313     DECLARE @end_date DATE = '2024-12-05';
314
315     INSERT INTO bookings (room_id, booker_id, start_date, end_date, status_id)
316     VALUES (@room_id, @client_id, @start_date, @end_date, 1);
317
318     IF (@end_date < @start_date)
319     BEGIN
320         RAISERROR('Invalid booking period: end_date is earlier than start_date.', 16, 1);
321         ROLLBACK TRAN InsertInvalidBookingDates;
322         RETURN;
323     END
324
325     COMMIT TRAN InsertInvalidBookingDates;
326 END TRY
327 BEGIN CATCH
328     ROLLBACK TRAN InsertInvalidBookingDates;
329     PRINT 'Error in InsertInvalidBookingDates transaction: ' + ERROR_MESSAGE();
330 END CATCH
331 GO
332
333 SELECT * FROM bookings WHERE room_id = 5;
334
```

Results Messages

id	room_id	booker_id	start_date	end_date	status_id	booking_date
5	5	JUR005	2024-11-09	2024-11-14	4	2024-11-26

```
-- 30.2 Транзакция добавления бронирования с end_date < start_date
-- Описание запроса: Пытается добавить бронирование, где end_date раньше start_date, проверяет логические ограничения.
BEGIN TRAN InsertInvalidBookingDates
BEGIN TRY
    DECLARE @room_id INT = 5;
    DECLARE @client_id NVARCHAR(255) = 'EF2345678';
    DECLARE @start_date DATE = '2024-12-10';
    DECLARE @end_date DATE = '2024-12-05';

    INSERT INTO bookings (room_id, booker_id, start_date, end_date, status_id)
    VALUES (@room_id, @client_id, @start_date, @end_date, 1);

    IF (@end_date < @start_date)
    BEGIN
        RAISERROR('Invalid booking period: end_date is earlier than start_date.', 16, 1);
        ROLLBACK TRAN InsertInvalidBookingDates;
        RETURN;
    END

    COMMIT TRAN InsertInvalidBookingDates;
END TRY
BEGIN CATCH
    ROLLBACK TRAN InsertInvalidBookingDates;
    PRINT 'Error in InsertInvalidBookingDates transaction: ' + ERROR_MESSAGE();
END CATCH
GO

SELECT * FROM bookings WHERE room_id = 5;
```

Messages

2:27:52 PM Started executing query at Line 308
(1 row affected)
Error in InsertInvalidBookingDates transaction: Invalid booking period: end_date is earlier than start_date.
Total execution time: 00:00:00.003