|  |  |
| --- | --- |
| **лого для документов 2022** | ***Федеральное агентство по рыболовству***  ***Федеральное государственное бюджетное образовательное***  ***учреждение высшего образования***  ***«Астраханский государственный технический университет»***  **Система менеджмента качества в области образования, воспитания, науки и инноваций**  **ООО «ДКС РУС» по международному стандарту ISO 9001:2015** |

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И КОММУНИКАЦИЙ

**КАФЕДРА АВТОМАТИЗИРОВАННЫХ СИСТЕМ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ**

**БАЗЫ ДАННЫХ**

(*наименование дисциплины*)

**ОТЧЕТ**

о выполнении заданий к лабораторной работе № 17-18

**Процедуры и триггеры**

|  |  |
| --- | --- |
|  | Выполнила:  студент гр. ДИНРБ-31 |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Кузургалиев Р.А. |
|  | «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_2023 г. |
|  | Максимальное количеству баллов \_\_\_\_\_  ЗАЩИЩЕНО:  Получено баллов \_\_\_\_\_\_ |
|  | Преподаватель: \_\_\_\_\_\_\_\_ Мамлеева А.Р. |
|  | «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2023 г. |

**Астрахань – 2023**

**Лабораторная работа №17-18 «Процедуры и триггеры»:**

1. Цель работы: разобраться в устройстве языка управления данных DML, а также в написании функций.

**ЗАДАНИЕ К ЛАБОРАТОРНОЙ РАБОТЕ №17-18**

Схема базы данных состоит из четырех отношений:

* Кассир
* Продукт
* Клиент
* Продажа

Схема базы данных представлена на рис. 1.

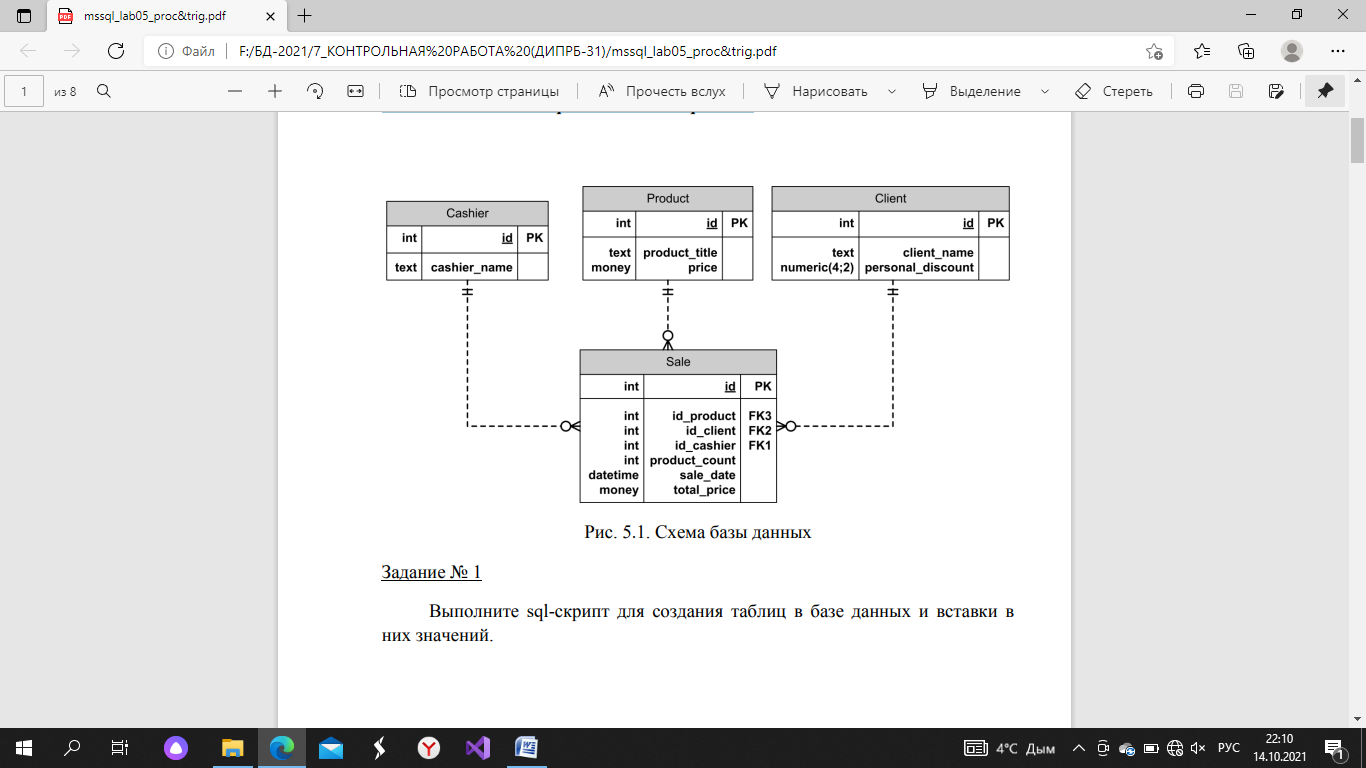


Рисунок 1 - Схема базы данных

**Задание № 1.** Выполните sql-скрипт для создания таблиц в базе данных и вставки в них значений.

CREATE TABLE Кассир (

id INT PRIMARY KEY,

имя NVARCHAR(50),

);

-- Вставка значений в таблицу "Кассир"

INSERT INTO Кассир (id, имя)

VALUES (1, N'Иван'),

(2, N'Анна'),

(3, N'Петр');

-- Создание таблицы "Продукт"

CREATE TABLE Продукт (

id INT PRIMARY KEY,

название NVARCHAR(50),

цена DECIMAL(10, 2),

);

-- Вставка значений в таблицу "Продукт"

INSERT INTO Продукт (id, название, цена)

VALUES (1, N'Молоко', 50.00),

(2, N'Хлеб', 30.00),

(3, N'Яйца', 70.00);

-- Создание таблицы "Клиент"

CREATE TABLE Клиент (

id INT PRIMARY KEY,

имя NVARCHAR(50),

скидка FLOAT,

);

-- Вставка значений в таблицу "Клиент"

INSERT INTO Клиент (id, имя, скидка)

VALUES (1, N'Мария', 0.2),

(2, N'Алексей',0.3),

(3, N'Елена', 0.4);

-- Создание таблицы "Продажа"

CREATE TABLE Продажа (

id INT PRIMARY KEY,

id\_кассира INT,

id\_продукта INT,

id\_клиента INT,

дата\_продажи DATE,

стоимость FLOAT,

FOREIGN KEY (id\_кассира) REFERENCES Кассир (id),

FOREIGN KEY (id\_продукта) REFERENCES Продукт (id),

FOREIGN KEY (id\_клиента) REFERENCES Клиент (id)

);

**Задание №2**. Создайте хранимую процедуру, которая устанавливает базовый уровень скидок для клиентов следующим образом:

* если клиент совершил не менее 5 покупок - 3%
* если сумма покупок превышает 50 000 - 5%

Выбирается максимальная скидка.

CREATE PROCEDURE УстановитьСкидку(

Id Int

)

BEGIN

UPDATE Клиент SET скидка = 0;

UPDATE Клиент SET скидка = 3

WHERE id IN (

SELECT id\_клиента

FROM Продажа

GROUP BY id\_клиента

HAVING COUNT(\*) >= 5

);

UPDATE Клиент SET скидка = 5

WHERE id IN (

SELECT id\_клиента

FROM Продажа

GROUP BY id\_клиента

HAVING SUM(цена) > 50000

);

END

EXEC УстановитьСкидку @id = 25;



**Задание №3.** Создайте хранимые процедуры для редактирования (добавление, удаление, изменение) значений всех справочников.

CREATE PROCEDURE AddCashier

@name NVARCHAR(50)

AS

BEGIN

INSERT INTO Кассир (имя) VALUES (@name)

END;

CREATE PROCEDURE DeleteCashier

@id INT

AS

BEGIN

DELETE FROM Кассир WHERE id = @id

END;

CREATE PROCEDURE UpdateCashier

@id INT,

@name NVARCHAR(50)

AS

BEGIN

UPDATE Кассир SET имя = @name WHERE id = @id

END;

CREATE PROCEDURE AddClient

@name NVARCHAR(50)

AS

BEGIN

INSERT INTO Клиент (имя) VALUES (@name)

END;

CREATE PROCEDURE DeleteClient

@id INT

AS

BEGIN

DELETE FROM Клиент WHERE id = @id

END;

CREATE PROCEDURE UpdateClient

@id INT,

@name NVARCHAR(50)

AS

BEGIN

UPDATE Клиент SET имя = @name WHERE id = @id

END;

CREATE PROCEDURE AddProduct

@name NVARCHAR(50)

@price INT

AS

BEGIN

INSERT INTO Продукт (название, цена) VALUES (@name, @price )

END;

CREATE PROCEDURE DeleteProduct

@id INT

AS

BEGIN

DELETE FROM Продукт WHERE id = @id

END;

CREATE PROCEDURE UpdateProduct

@id INT,

@name NVARCHAR(50)

AS

BEGIN

UPDATE Продукт SET название = @name WHERE id = @id

END;

**Задание №4.** Создайте хранимую процедуру, которая возвращает размер скидки клиента на текущий момент:

* базовую скидку
* скидку выходного дня - 3%
* скидку за объем покупки (при покупке на сумму более 100 000) - 1%

Скидки суммируются.

CREATE PROCEDURE GetClientDiscount

@client\_id INT,

@current\_date DATE

AS

BEGIN

DECLARE @base\_discount DECIMAL(5,2)

DECLARE @weekend\_discount DECIMAL(5,2)

DECLARE @volume\_discount DECIMAL(5,2)

SELECT @base\_discount = скидка FROM Клиент WHERE id = @client\_id

IF DATENAME(dw, @current\_date) IN ('Saturday', 'Sunday')

SET @weekend\_discount = 0.03

ELSE

SET @weekend\_discount = 0

SELECT @volume\_discount = IIF(SUM(стоимость) > 100000, 0.01, 0)

FROM Продажа

WHERE id\_клиента = @client\_id

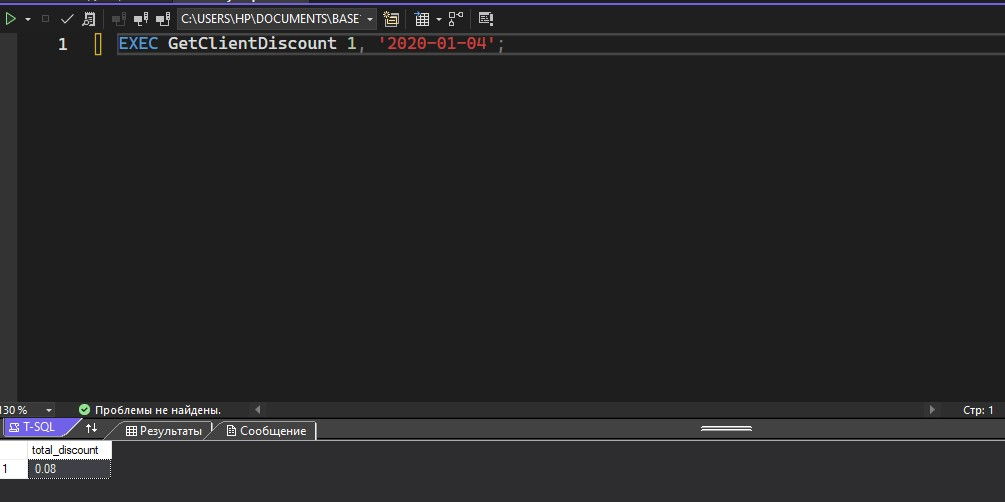
AND MONTH(дата\_продажи) = MONTH(@current\_date)

AND YEAR(дата\_продажи) = YEAR(@current\_date)

SELECT (@base\_discount + @weekend\_discount + @volume\_discount) AS total\_discount

END

EXEC GetClientDiscount 1, '2020-01-04';



**Задание №5.** Создайте триггер на изменение таблицы продаж (на вставку) для расчета общей суммы продажи на текущий момент.

Синтаксис создания **DML-триггера** следующий:

CREATE TRIGGER расчет\_общей\_суммы

ON Продажа

AFTER INSERT

AS

BEGIN

-- Обновление общей суммы продажи на текущий момент

UPDATE Продажа

SET общая\_сумма = (

SELECT SUM(стоимость)

FROM inserted

WHERE Продажа.id\_кассира = inserted.id\_кассира

)

FROM Продажа

INNER JOIN inserted ON Продажа.id = inserted.id;

END;

**Задание №6**. Создайте триггер на удаление строки из таблицы продаж (на удаление), который обеспечит пересчѐт базового уровня скидки клиента.

CREATE TRIGGER UpdateDiscount

ON Продажа

AFTER DELETE

AS

BEGIN

DECLARE @clientId INT

SELECT @clientId = id\_клиента FROM deleted

UPDATE Клиент

SET скидка = (SELECT COUNT(\*) FROM Продажа WHERE id\_клиента = @clientId) \* 0.05

WHERE id = @clientId

END

**Задание №7.** Создайте хранимую процедуру, которая принимает в качестве параметра id клиента и возвращает нарастающим итогом (дата, общая сумма) информацию обо всех покупках клиента.

CREATE PROCEDURE GetClientSales

@clientId INT

AS

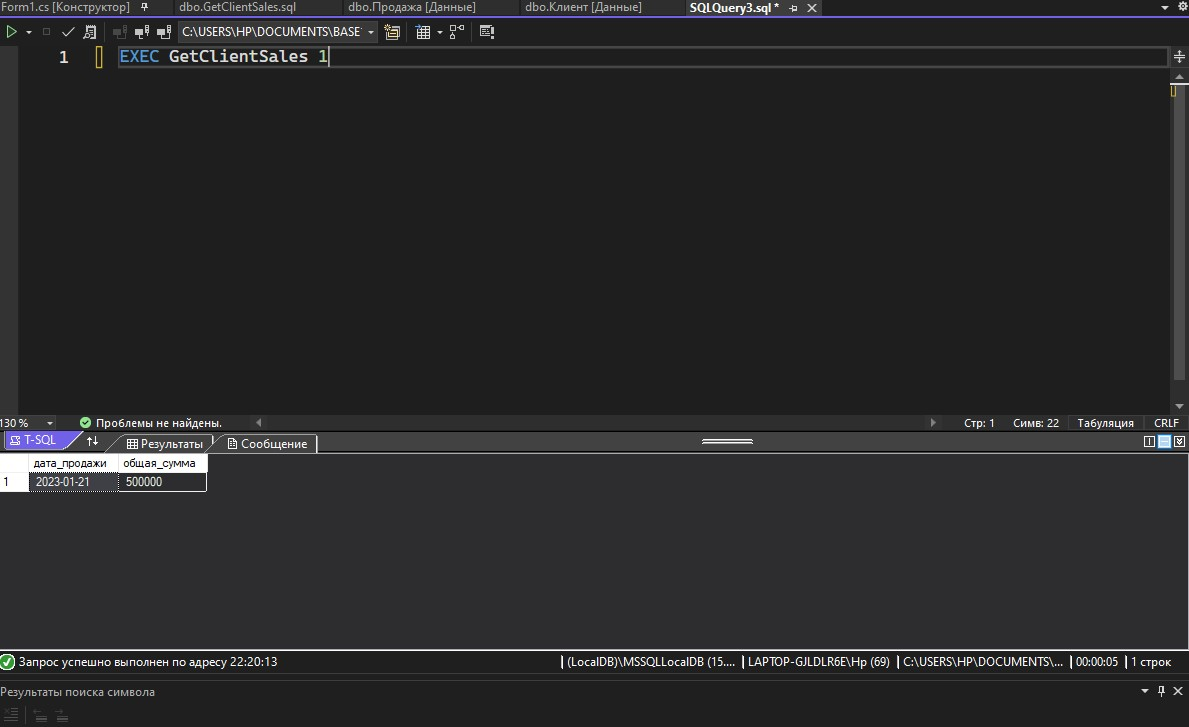
BEGIN

SELECT дата\_продажи, SUM(стоимость) OVER (ORDER BY дата\_продажи) AS общая\_сумма

FROM Продажа

WHERE id\_клиента = @clientId

END



**Задание №8.** Создайте хранимую процедуру, возвращающую значение бонуса кассиру (id кассира) по месяцам, в которых у него были продажи.

Размер бонуса составляет:

* 0.1% от суммы продажи, если 10 000 > сделка >
* 0 0.2% от суммы продажи, если 50 0000 > сделка > 10 000
* 0.3% от суммы продажи, если 100 0000 > сделка > 50 000

CREATE PROCEDURE GetCashierBonusByMonth

@cashierId INT

AS

BEGIN

SELECT MONTH(дата\_продажи) AS месяц,

SUM(

CASE

WHEN Продажа.стоимость < 10000 THEN Продажа.стоимость \* 0.001

WHEN Продажа.стоимость >= 10000 AND Продажа.стоимость < 50000 THEN Продажа.стоимость \* 0.002

WHEN Продажа.стоимость >= 50000 AND Продажа.стоимость < 100000 THEN Продажа.стоимость \* 0.003

ELSE 0

END

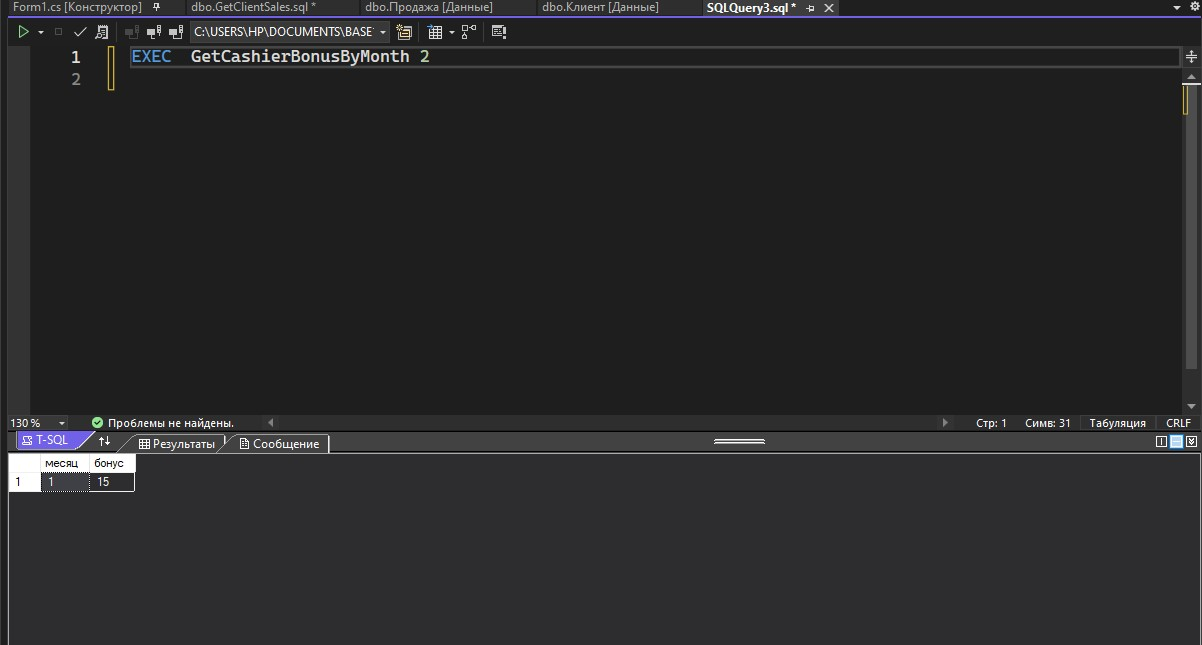
) AS бонус

FROM Продажа

WHERE id\_кассира = @cashierId

GROUP BY MONTH(дата\_продажи)

END



**Задание №9.** Создайте триггер, который запрещает удаление созданных вами пользовательских таблиц.

Синтаксис создания **DDL-триггера** следующий:

CREATE TRIGGER PreventTableDeletion

ON DATABASE

FOR DROP\_TABLE

AS

BEGIN

DECLARE @TableName NVARCHAR(100)

SET @TableName = EVENTDATA().value('(/EVENT\_INSTANCE/ObjectName)[1]', 'NVARCHAR(100)')

IF @TableName IN ('Кассир', 'Клиент', 'Продукт', 'Продажа')

BEGIN

RAISERROR('Cannot delete user-created table %s', 16, 1, @TableName)

ROLLBACK

END

END