Федеральное государственное образовательное бюджетное учреждение высшего образования

«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ

РОССИЙСКОЙ ФЕДЕРАЦИИ»

**Колледж информатики и программирования**

Работа по ТРЗБД

по теме: «**Разработка базы данных для управления онлайн-магазином**»

Выполнили:

студент группы 3ИСИП-422

Сидаков Амир Азаматович

Проверил:

преподаватель

Тарасов И. Ю.

«7»декабрь 2024г.

Москва

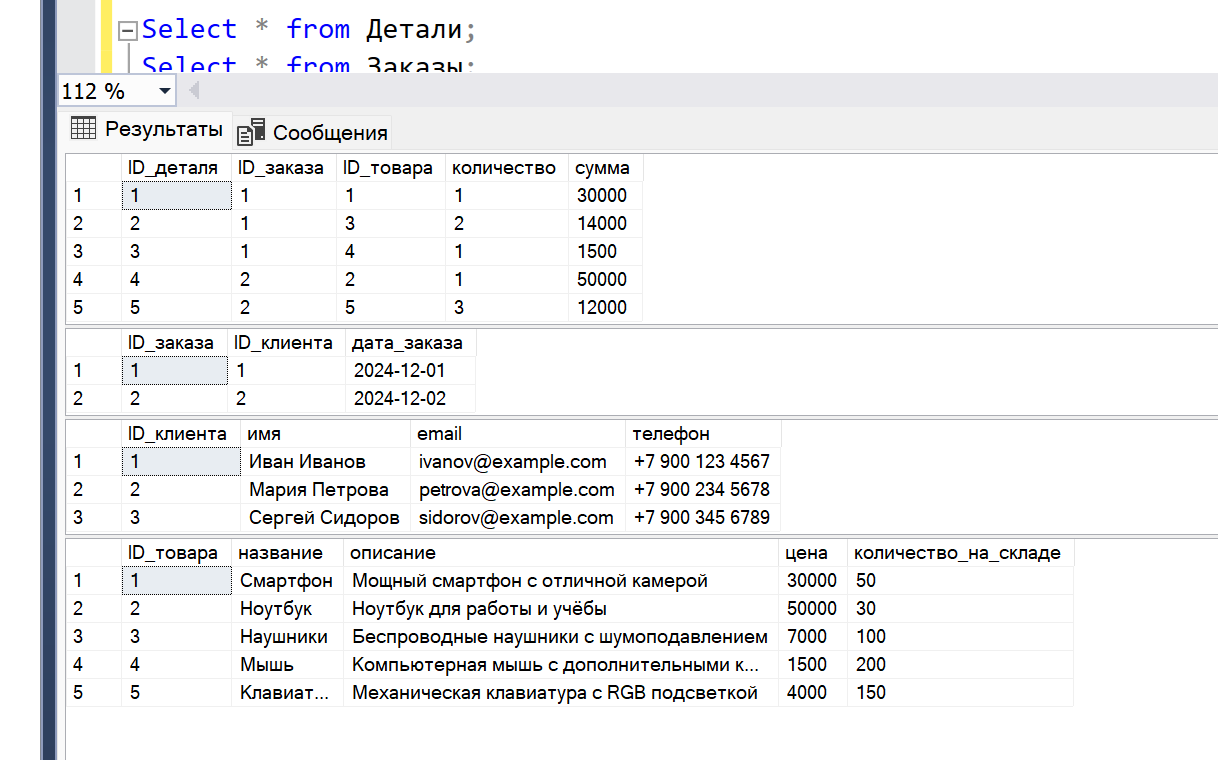
2024

**Цель:** спроектировать и создать базу данных, которая поддерживает управление товарами, клиентами, заказами и отправкой уведомлений клиентам.

**Задание:**

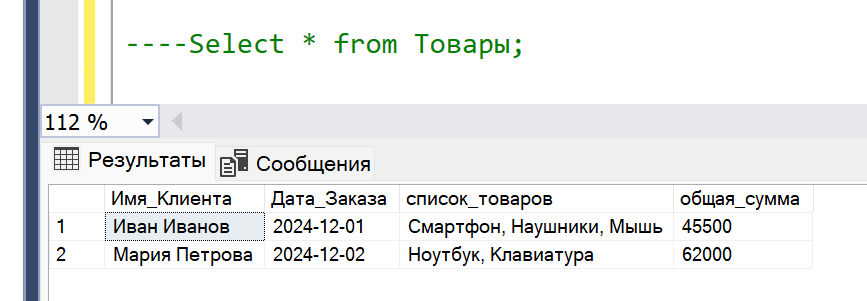
1. **Создание структуры базы данных:**
   * Создайте таблицы для хранения информации о:
     + Клиентах (ID клиента, имя, email, телефон).
     + Товарах (ID товара, название, описание, цена, количество на складе).
     + Заказах (ID заказа, ID клиента, дата заказа).
     + Деталях заказа (ID детали, ID заказа, ID товара, количество, сумма).
2. **Добавление данных:**
   * Вставьте данные о 3 клиентах.
   * Добавьте 5 товаров с различными характеристиками.
   * Создайте 2 заказа с деталями.

|  |
| --- |
| 1. --create table Клиенты 2. --( 3. -- ID\_клиента int primary key not null, 4. -- имя varchar(40), 5. -- email varchar(40), 6. -- телефон varchar(40) 7. --) 8. --create table Товары 9. --( 10. -- ID\_товара int primary key not null, 11. -- название varchar(40), 12. -- описание text, 13. -- цена int, 14. -- количество\_на\_складе int 15. --) 16. --create table Заказы 17. --( 18. -- ID\_заказа int primary key not null, 19. -- ID\_клиента int, 20. -- дата\_заказа date, 21. -- foreign key (ID\_клиента) references Клиенты(ID\_клиента) 22. --) 23. --create table Детали 24. --( 25. -- ID\_деталя int primary key not null, 26. -- ID\_заказа int, 27. -- ID\_товара int, 28. -- количество int, 29. -- сумма int, 30. -- foreign key (ID\_заказа) references Заказы(ID\_заказа), 31. -- foreign key (ID\_товара) references Товары(ID\_товара) 32. --) 33. ------ Вставляем 3 клиентов 34. ----INSERT INTO Клиенты (ID\_клиента, имя, email, телефон) 35. ----VALUES 36. ---- (1, 'Иван Иванов', 'ivanov@example.com', '+7 900 123 4567'), 37. ---- (2, 'Мария Петрова', 'petrova@example.com', '+7 900 234 5678'), 38. ---- (3, 'Сергей Сидоров', 'sidorov@example.com', '+7 900 345 6789'); 39. ------ Вставляем 5 товаров 40. ----INSERT INTO Товары (ID\_товара, название, описание, цена, количество\_на\_складе) 41. ----VALUES 42. ---- (1, 'Смартфон', 'Мощный смартфон с отличной камерой', 30000, 50), 43. ---- (2, 'Ноутбук', 'Ноутбук для работы и учёбы', 50000, 30), 44. ---- (3, 'Наушники', 'Беспроводные наушники с шумоподавлением', 7000, 100), 45. ---- (4, 'Мышь', 'Компьютерная мышь с дополнительными кнопками', 1500, 200), 46. ---- (5, 'Клавиатура', 'Механическая клавиатура с RGB подсветкой', 4000, 150); 47. ------ Вставляем 2 заказа 48. ----INSERT INTO Заказы (ID\_заказа, ID\_клиента, дата\_заказа) 49. ----VALUES 50. ---- (1, 1, '2024-12-01'), -- Заказ 1 от клиента 1 (Иван Иванов) 51. ---- (2, 2, '2024-12-02'); -- Заказ 2 от клиента 2 (Мария Петрова) 52. -- Вставка данных в таблицу Детали для Заказа 1 (Иван Иванов) 53. --INSERT INTO Детали (ID\_деталя, ID\_заказа, ID\_товара, количество, сумма) 54. --VALUES 55. -- (1, 1, 1, 1, 30000), 56. -- (2, 1, 3, 2, 14000), 57. -- (3, 1, 4, 1, 1500), 58. -- (4, 2, 2, 1, 50000), 59. -- (5, 2, 5, 3, 12000); 60. Select \* from Детали; 61. Select \* from Заказы; 62. Select \* from Клиенты; 63. Select \* from Товары; |



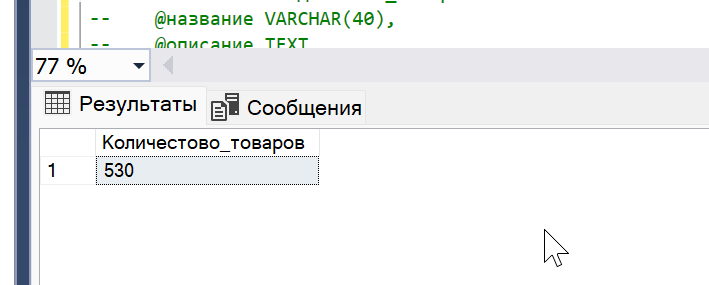
1. **Создание представлений:**
   * Создайте представление для получения информации о заказах (имя клиента, дата заказа, список товаров, общая сумма заказа).

|  |
| --- |
| 1. --create view Инфо\_о\_заказах as 2. --select 3. -- К.имя as Имя\_Клиента, 4. -- З.дата\_заказа as Дата\_Заказа, 5. -- STRING\_AGG(Т.название, ', ') AS список\_товаров, 6. -- SUM(Д.сумма) AS общая\_сумма 7. --from Заказы З 8. --left join Клиенты К 9. --on З.ID\_клиента = К.ID\_клиента 10. --left join Детали Д 11. --on З.ID\_заказа = Д.ID\_заказа 12. --left join Товары Т 13. --on Д.ID\_товара = Т.ID\_товара 14. --group by 15. -- К.имя, З.дата\_заказа 17. SELECT \* FROM Инфо\_о\_заказах; |



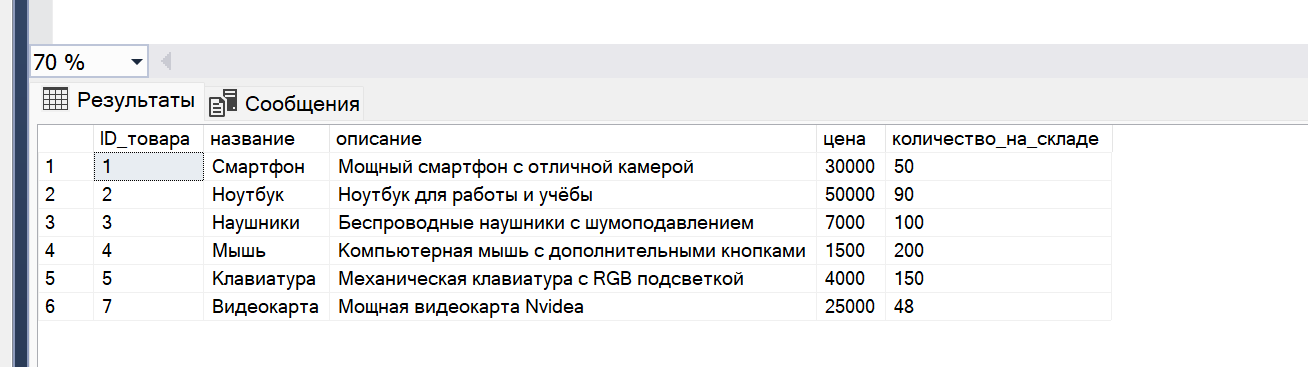
1. **Создание функций:**
   1. Напишите функцию, которая возвращает общее количество товаров на складе.

|  |
| --- |
| --create function TotalItems()  --returns int  --as  --begin  -- declare @totalItems int;  -- Set @totalItems = (select SUM(количество\_на\_складе) from Товары);  -- return @totalItems;  --end;  select [dbo].[TotalItems]() as Количестово\_товаров |



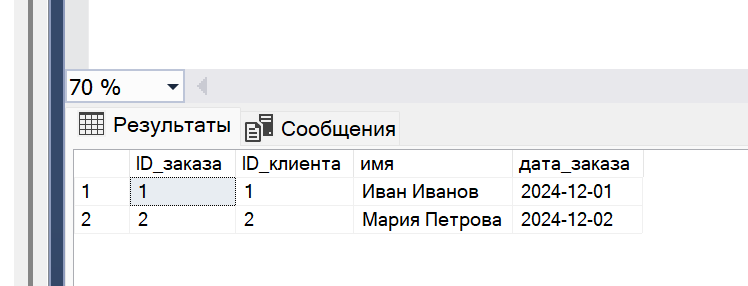
1. **Создание процедуры:**
   1. Напишите хранимую процедуру для добавления нового товара в таблицу с проверкой: если товар с таким названием уже существует, обновите его количество на складе.

|  |
| --- |
| 1. --CREATE PROCEDURE Добавить\_товар 2. -- @название VARCHAR(40), 3. -- @описание TEXT, 4. -- @цена INT, 5. -- @количество INT 6. --AS 7. --BEGIN 8. -- --Проверка, существует ли товар с таким названием 9. -- IF EXISTS (SELECT 1 FROM Товары WHERE название = @название) 10. -- BEGIN 11. -- --Если товар существует, обновляем его количество 12. -- UPDATE Товары 13. -- SET количество\_на\_складе = количество\_на\_складе + @количество 14. -- WHERE название = @название; 15. -- END 16. -- ELSE 17. -- BEGIN 18. -- --Если товара нет, вставляем новый товар 19. -- declare @ID\_товара int; 20. -- INSERT INTO Товары (ID\_товара, название, описание, цена, количество\_на\_складе) 21. -- VALUES (@ID\_товара, @название, @описание, @цена, @количество); 22. -- END 23. --END; 24. -- Вставляем новый товар 25. EXEC Добавить\_товар 'Видеокарта', 'Мощная видеокарта Nvidea', 25000, 3; 26. -- Обновляем количество существующего товара (например, 'Ноутбук') 27. EXEC Добавить\_товар 'Ноутбук', 'Ноутбук для работы и учёбы', 50000, 5; 28. SELECT \* FROM Товары; |



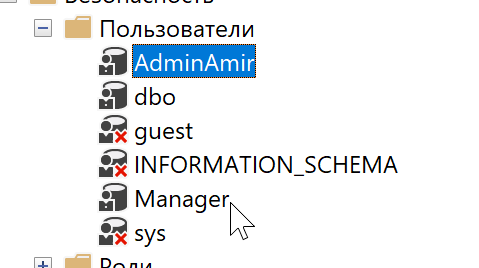
1. **Использование временных таблиц:**
   1. Создайте временную таблицу, в которую загрузите список клиентов, сделавших хотя бы один заказ.

|  |
| --- |
| 1. --create table #ListOfClients 2. --( 3. -- ID\_заказа int, 4. -- ID\_клиента int, 5. -- имя varchar(40), 6. -- дата\_заказа date 7. --) 8. select 9. З.ID\_заказа, 10. К.ID\_клиента, 11. К.имя, 12. З.дата\_заказа 13. into #ListOfClients from Заказы З 14. inner join Клиенты К 15. on З.ID\_клиента = К.ID\_клиента; 16. select \* from #ListOfClients; |

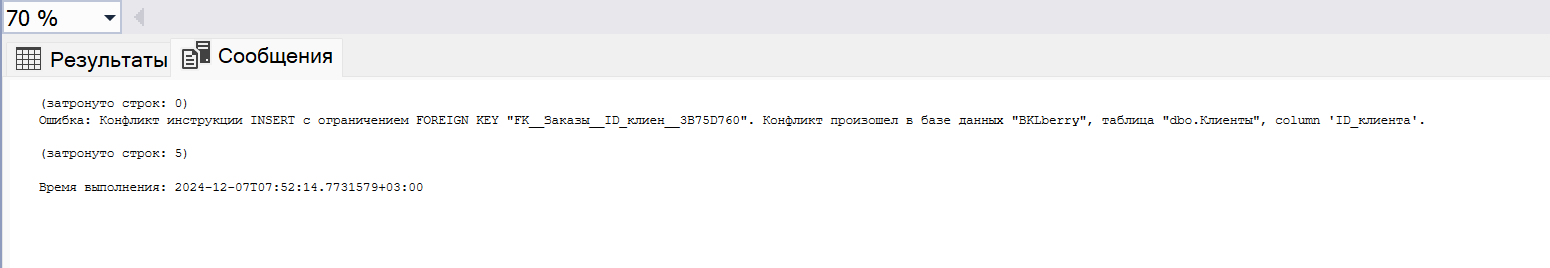


1. **Контроль данных:**
   * Настройте доступ к таблицам:
     + Пользователю *Manager* разрешите только чтение данных о заказах.
     + Пользователю *Admin* разрешите полный доступ к таблицам.
   * Используйте транзакцию, чтобы добавить заказ с деталями. Если одна из операций вставки деталей заказа завершится ошибкой, отмените транзакцию.

|  |
| --- |
| 1. CREATE LOGIN Logins WITH PASSWORD = 'MyPass'; 2. create user Manager for login Logins; 3. grant select on [dbo].[Заказы] to Manager; 4. CREATE LOGIN LoginAdmin WITH PASSWORD = 'MyPassAdmin'; 5. create user AdminAmir for login LoginAdmin; 6. grant control on [dbo].[Заказы] to Manager; |



|  |
| --- |
| --CREATE LOGIN Logins WITH PASSWORD = 'MyPass';  --create user Manager for login Logins;  --grant select on [dbo].[Заказы] to Manager;  --CREATE LOGIN LoginAdmin WITH PASSWORD = 'MyPassAdmin';  --create user AdminAmir for login LoginAdmin;  --grant control on [dbo].[Заказы] to Manager;  begin transaction;  begin try  INSERT INTO Заказы (ID\_заказа, ID\_клиента, дата\_заказа)  VALUES  (5, 5, '2024-12-03'); -- Заказ от клиента 4 (Сергей Сидоров)  -- Шаг 2: Вставка деталей заказа в таблицу Детали  INSERT INTO Детали (ID\_деталя, ID\_заказа, ID\_товара, количество, сумма)  VALUES  (10, 5, 10, 2, 6000), -- Смартфоны  (11, 5, 11, 1, 7000), -- Наушники  (12, 5, 12, 2, 3000) -- Мышь  -- Если всё прошло успешно, завершаем транзакцию  COMMIT TRANSACTION;  end try  begin catch  -- Если произошла ошибка, откатываем транзакцию  ROLLBACK TRANSACTION;  -- Выводим сообщение об ошибке  PRINT 'Ошибка: ' + ERROR\_MESSAGE();  end catch;  select \* from Заказы  join Детали  on Заказы.ID\_заказа = Детали.ID\_заказа |



1. **Почтовая рассылка:**
   1. Настройте отправку уведомления клиенту о его заказе, используя встроенную функцию отправки email в SQL Server.

|  |
| --- |
| --CREATE PROCEDURE ОтправитьУведомлениеОЗаказе  -- @ID\_заказа INT  --AS  --BEGIN  -- DECLARE @имя NVARCHAR(40),  -- @email NVARCHAR(40),  -- @детали NVARCHAR(MAX),  -- @сумма INT,  -- @тема NVARCHAR(100),  -- @сообщение NVARCHAR(MAX);  -- -- Получаем информацию о клиенте и заказе  -- SELECT  -- @имя = К.имя,  -- @email = К.email  -- FROM Заказы З  -- JOIN Клиенты К ON З.ID\_клиента = К.ID\_клиента  -- WHERE З.ID\_заказа = @ID\_заказа;  -- -- Получаем детали заказа и общую сумму  -- SELECT  -- @детали = STRING\_AGG(Т.название + ' (количество: ' + CAST(Д.количество AS NVARCHAR) + ')', ', '),  -- @сумма = SUM(Д.сумма)  -- FROM Детали Д  -- JOIN Товары Т ON Д.ID\_товара = Т.ID\_товара  -- WHERE Д.ID\_заказа = @ID\_заказа;  -- -- Формируем тему и сообщение  -- SET @тема = 'Ваш заказ №' + CAST(@ID\_заказа AS NVARCHAR);  -- SET @сообщение = 'Уважаемый(ая) ' + @имя + ',' + CHAR(13) + CHAR(10) +  -- 'Ваш заказ успешно оформлен.' + CHAR(13) + CHAR(10) +  -- 'Детали заказа: ' + @детали + CHAR(13) + CHAR(10) +  -- 'Общая сумма: ' + CAST(@сумма AS NVARCHAR) + ' руб.' + CHAR(13) + CHAR(10) +  -- 'Спасибо за ваш выбор!';  -- -- Отправляем письмо  -- EXEC msdb.dbo.sp\_send\_dbmail  -- @profile\_name = 'amir.sidackov@yandex.ru', -- Укажите имя профиля Database Mail  -- @recipients = @email,  -- @subject = @тема,  -- @body = @сообщение;  -- PRINT 'Уведомление отправлено успешно.';  --END;  exec ОтправитьУведомлениеОЗаказе @ID\_заказа = 2;  --drop proc ОтправитьУведомлениеОЗаказе |

1. Результат процедуры: Первое – показываю что пришло в общем. Второе – что будет если в @recipients указать не мой Email и Третье - @recipients = мой Email.

