--Создание таблиц

CREATE TABLE Sellers

(

[SellerId] int identity primary key,

[Name] nvarchar(200),

);

GO

CREATE TABLE Products

(

[ProductId] int identity primary key,

[Name] nvarchar(500),

[SellerId] int,

[Rating] decimal(2,1),

[Price] money,

[Аmount] int,

CONSTRAINT [FK\_Products\_EmployeeId] FOREIGN KEY (SellerId) REFERENCES Sellers(SellerId)

);

GO

CREATE TABLE [Pick-up points]

(

[Pick-up-pointId] int identity primary key,

[Address] nvarchar(300),

[Rating] decimal(2,1),

[Orders for last month] int

);

GO

CREATE TABLE [Employees]

(

[EmployeeId] int identity primary key,

[Name] nvarchar(200),

[Role] nvarchar(200),

[Password] nvarchar(200),

[Pick-up-pointId] int,

[Salary] money

CONSTRAINT [FK\_Employees\_Pick-up point Id] FOREIGN KEY ([Pick-up-pointId]) REFERENCES [Pick-up points]([Pick-up-pointId])

);

GO

CREATE TABLE Clients

(

[ClientId] int identity primary key,

[Name] nvarchar(200),

[Surname] nvarchar(200),

[Password] nvarchar(200),

[Email] nvarchar(70)

);

GO

CREATE TABLE Orders

(

[OrderId] int identity primary key,

[ClientId] int,

[EmployeeId] int,

[Cost] money,

CONSTRAINT [FK\_ClientId\_Clients] FOREIGN KEY (ClientId) REFERENCES Clients(ClientId) ON DELETE SET NULL,

CONSTRAINT [FK\_EmployeeId\_Employees] FOREIGN KEY (EmployeeId) REFERENCES Employees(EmployeeId) ON DELETE SET NULL

);

GO

CREATE TABLE [Orders\_Products]

(

[Order\_ProductId] int identity primary key,

[OrderId] int,

[ProductId] int,

[Amount] int,

CONSTRAINT [FK\_Orders\_Products\_ProductId] FOREIGN KEY (ProductId) REFERENCES Products(ProductId),

CONSTRAINT [FK\_Orders\_Products\_OrderId] FOREIGN KEY (OrderId) REFERENCES Orders(OrderId)

);

--Заполнение таблиц

INSERT INTO [Products] ([Name], [Price], [Rating], [SellerId], [Аmount])

VALUES

('Смартфон POCQ X5 5G global edition', 7850, 5.0, 1, 1),

('Смартфон Redme Note 12S сеть 5G', 3875, 5.0, 2, 1),

('Смартфон Camon 20 Pro Русская версия Сеть 5G', 7850, 3.0, 3, 1),

('Мороженное за 3 копейки', 0.3, 1.3, 4, 1),

('Смартфон POCQ X5 5G global edition', 3819, 4.0, 5, 1),

('Смартфон Xioami 14 Pro', 3605, 3.5, 1, 1),

('Смартфон Мобильный телефон Xaomi 13T Pro Ultimate 5G 16/512 ГБ', 7887, 5.0, 2, 1),

('Poco Смартфон C65 8/256 ГБ, черный', 10904, 5.0, 3, 1),

('Смартфон realmei C53 поддерживает Google,NFC', 8201, 5.0, 4, 1),

('Infinix Смартфон NOTE 30 X6833B 8/256 ГБ, голубой', 13875, 5.0, 5, 1),

('Смартфон Spark 10 Pro поддерживает быструю зарядку', 3785, 5.0, 1, 1),

('Tecno Смартфон Spark 20 8/128 ГБ', 10227, 3.7, 2, 1),

('Беспроводные наушники Samsung Air', 1373, 2.7, 3, 1),

('Samsung Смартфон Galaxy S21 8/256 ГБ', 30875, 2.0, 4, 1),

('Samsung Умные часы Galaxy Watch6, 47mm', 27850, 5.0, 5, 1),

('Samsung Смартфон Samsung Galaxy A54 5G 8/128 ГБ', 35875, 5.0, 1, 1),

('Блок питания с проводом Samsung 15W PD Power Adapter USB', 1050, 5.0, 2, 1),

('Samsung Телевизор UE43CU7100UXCE 43" 4K UHD', 32875, 5.0, 3, 1),

('Бритва жилет', 350, 5.0, 4, 1),

('Носки прада', 300875, 1.5, 5, 1);

GO

INSERT INTO [Pick-up points] ([Address], [Rating], [Orders for last month])

VALUES ('123 Main St', 4.5, 10),

('456 Elm St', 3.4, 5),

('789 Oak St', 4.2, 8),

('321 Pine St', 4.0, 0),

('654 Maple St', 4.7, 12);

GO

INSERT INTO [Employees] ([Name], [Role], [Password], [Pick-up-pointId], [Salary])

VALUES ('John', 'Administrator', '1', 1, 5000),

('Jane', 'Administrator', '1', 2, 6000),

('Michael', 'Loader', '1', 3, 5500),

('Emily', 'Loader', '1', 4, 5200),

('David', 'Loader', '1', 5, 5800);

INSERT INTO Clients ([Name], [Surname], [Password], [Email])

VALUES ('John', 'Doe', '1', 'johndoe@example.com'),

('Jane', 'Smith', '1', 'janesmith@example.com'),

('Michael', 'Brown', '1', 'michaelbrown@example.com'),

('Emily', 'Johnson', '1', 'emilyjohnson@example.com'),

('David', 'Miller', '1', 'davidmiller@example.com');

INSERT INTO Orders ([ClientId], [EmployeeId], [Cost])

VALUES (1, 1, 39250),

(2, 2, 11625),

(3, 3, 15700),

(4, 4, 111400),

(5, 5, 350);

INSERT INTO Sellers ([SellerId], [Name])

VALUES (1, Aliexpress),

(2, Microsoft),

(3, True Engineering),

(4, Take two),

(5, Coca cola industries);

INSERT INTO [Orders\_Products] ([OrderId], [ProductId], [Amount])

VALUES (1, 7, 5),

(2, 9, 3),

(3, 5, 2),

(4, 15, 4),

(5, 19, 1);

--Запросы

-- 1

GO

CREATE PROCEDURE GetProductsFromSeller AS

SELECT [Name], [Rating], [Price]

FROM [Products]

WHERE [SellerId] = 1;

-- 2

GO

CREATE PROCEDURE SearchProductByName AS

SELECT [Name], [Rating], [Price]

FROM [Products]

WHERE LOWER([Name]) LIKE '%носки%';

-- 3

GO

CREATE PROCEDURE ShowProductsByRating AS

SELECT [Name], [Rating], [Price]

FROM [Products]

ORDER BY [Rating];

-- 4

GO

CREATE PROCEDURE From1To3 AS

SELECT \* FROM

(SELECT \* FROM (SELECT [Name], [Rating], [Price] FROM [Products]

WHERE [SellerId] = 1) AS ProdFromSel

WHERE LOWER([Name]) LIKE '%носки%') AS ProdWithName

ORDER BY [Rating];

-- 5

GO

CREATE PROCEDURE LikeFourButSkip AS

SELECT \* FROM

(SELECT \* FROM (SELECT [Name], [Rating], [Price] FROM [Products]

WHERE [SellerId] = 1) AS ProdFromSel

WHERE LOWER([Name]) LIKE '%носки%') AS ProdWithName

ORDER BY [Rating]

OFFSET 2 ROWS FETCH NEXT 2 ROWS ONLY;

-- 6

GO

CREATE PROCEDURE ProductsInExpOrder AS

SELECT \*

FROM (SELECT Products.[Name] AS [Name], Orders\_Products.[Amount] AS Amount

FROM Orders\_Products JOIN Products ON Orders\_Products.ProductId = Products.ProductId

WHERE OrderId = (SELECT TOP 1 [OrderId]

FROM [Orders]

ORDER BY [Cost] DESC)) AS List;

-- 7

GO

CREATE PROCEDURE ClientsWithMinOrders AS

SELECT [Name], [Surname], [Patronymic], [Email]

FROM [Orders] JOIN [Clients] ON [Orders].ClientId = [Clients].ClientId

WHERE [Orders].[OrderId] IN

(SELECT TOP 4 [OrderId]

FROM [Orders\_Products]

GROUP BY [OrderId]

ORDER BY SUM(Amount));

-- 8

GO

CREATE PROCEDURE IssuedOrdersByPoint AS

SELECT

[Emp\_clientInfo].[Employee's name],

[Emp\_clientInfo].[Employee's surname],

[Emp\_clientInfo].[Employee's patronymic],

[Emp\_clientInfo].[Client's name],

[Emp\_clientInfo].[Client's surname],

[Emp\_clientInfo].[Client's patronymic]

FROM (

SELECT

\*,

[Clients].[Name] AS [Client's name],

[Clients].[Surname] AS [Client's surname],

[Clients].[Patronymic] AS [Client's patronymic]

FROM

[Clients]

JOIN (

SELECT

[Name] AS [Employee's name],

[Surname] AS [Employee's surname],

[Patronymic] AS [Employee's patronymic],

[ClientId],

[Pick-up-pointId],

E.EmployeeId AS EmpId

FROM

[Orders] AS O

JOIN

[Employees] AS E ON O.EmployeeId = E.EmployeeId

) AS EmpInfo ON EmpInfo.ClientId = Clients.ClientId

) AS Emp\_clientInfo

WHERE

[Emp\_clientInfo].[Pick-up-pointId] = 2

GROUP BY

EmpId,

[Emp\_clientInfo].[Employee's name],

[Emp\_clientInfo].[Employee's surname],

[Emp\_clientInfo].[Employee's patronymic],

[Emp\_clientInfo].[Client's name],

[Emp\_clientInfo].[Client's surname],

[Emp\_clientInfo].[Client's patronymic];

-- 9

GO

CREATE PROCEDURE SalaryDecrease AS

UPDATE [Employees]

SET [Salary] = Salary \* 0.8

WHERE [Employees].[Pick-up-pointId] IN (SELECT \* FROM (SELECT [Pick-up points].[Pick-up-pointsId]

FROM [Employees] JOIN [Pick-up-points] ON [Employees].[Pick-up-pointId] = [Pick-up points].[Pick-up-pointsId]

WHERE [Rating] < 3.5) AS S);

-- 10

GO

CREATE PROCEDURE DeletePoints AS

DELETE [Pick-up points]

WHERE [Orders for last month] < 1;