**3.** Сделать запросы (+ балл, если это будут процедуры) на:

1. Получение списка товаров одного продавца;

CREATE PROCEDURE GetMarketProducts

@marketId INT = 3

AS

BEGIN

SELECT m.market\_id, m.name, pri.name

FROM Products\_Instances pri

JOIN Products p ON pri.product\_instance\_id = p.product\_instance\_id

JOIN Markets m ON m.market\_id = p.market\_id

WHERE m.market\_id = @marketId;

END;

1. Поиск определенного товара в списке (например, «носки» или «пюре»). Запрос должен вернуть все записи, содержащие искомую строку в любом месте названия;

CREATE PROCEDURE GetProductInstancesByName

@partialName NVARCHAR(100) = ‘by’

AS

BEGIN

SELECT \*

FROM Products\_Instances

WHERE [name] LIKE '%' + @partialName + '%';

END;

1. Вывод всех товаров с сортировкой по возрастанию рейтинга (от худшего к лучшему);

CREATE PROCEDURE GetProductInstancesWithRating

AS

BEGIN

SELECT pi.[name], p.rating

FROM Products p

JOIN Products\_Instances pi ON p.product\_instance\_id = pi.product\_instance\_id

ORDER BY p.rating ASC;

END;

1. Запрос, объединяющий все предыдущие;

CREATE PROCEDURE GetProductInstancesByMarketAndName

@marketId INT = 3,

@partialName NVARCHAR(100) = ‘by’

AS

BEGIN

SELECT m.market\_id, m.name, pri.name, p.rating

FROM Products\_Instances pri

JOIN Products p ON pri.product\_instance\_id = p.product\_instance\_id

JOIN Markets m ON m.market\_id = p.market\_id

WHERE m.market\_id = @marketId AND pri.name LIKE '%' + @partialName + '%';

END;

1. То же самое, что и 4, но только 2 записи, начиная с 3-ей;

CREATE PROCEDURE GetFilteredProductInstances

@marketId INT = 3,

@partialName NVARCHAR(100) = ‘’,

@offsetRows INT = 2,

@fetchRows INT = 3

AS

BEGIN

SELECT m.market\_id, m.name, pri.name, p.rating

FROM Products\_Instances pri

JOIN Products p ON pri.product\_instance\_id = p.product\_instance\_id

JOIN Markets m ON m.market\_id = p.market\_id

WHERE m.market\_id = @marketId AND pri.name LIKE '%' + @partialName + '%'

ORDER BY p.rating ASC

OFFSET @offsetRows ROWS

FETCH NEXT @fetchRows ROWS ONLY;

END;

1. Вывод списка товаров самого дорогого заказа, включая их количество, стоимость и поставщика (название, не ID);

CREATE PROCEDURE GetExpensiveOrderDetails

AS

BEGIN

DECLARE @expensiveOrderId INT;

SELECT TOP 1 @expensiveOrderId = order\_id

FROM Orders

ORDER BY total\_amount DESC;

SELECT OI.order\_id, P.price, PI.name AS product\_name, M.name AS market\_name, COUNT(\*) [count]

FROM Orders\_Items AS OI

JOIN Products AS P ON OI.product\_id = P.product\_id

JOIN Products\_Instances AS PI ON P.product\_instance\_id = PI.product\_instance\_id

JOIN Markets AS M ON P.market\_id = M.market\_id

WHERE OI.order\_id = @expensiveOrderId

GROUP BY OI.order\_id, P.price, PI.name, M.name

END

1. Вывод информации о 4 клиентах, которые делают самые маленькие заказы (по количеству товаров в заказе вне зависимости от цены);

CREATE PROCEDURE GetSmallestOrderClients

AS

BEGIN

WITH OrderQuantities AS (

SELECT order\_id, COUNT(\*) AS num\_items

FROM Orders\_Items

GROUP BY order\_id

), SmallestOrders AS (

SELECT TOP 4 order\_id, num\_items

FROM OrderQuantities

ORDER BY num\_items

)

SELECT C.name, C.email, C.phone, SO.num\_items

FROM Clients C

JOIN Orders AS O ON C.client\_id = O.client\_id

JOIN SmallestOrders AS SO ON O.order\_id = SO.order\_id

ORDER BY SO.num\_items;

END;

1. Информация о работе ПВЗ: какой сотрудник обслужил какого клиента и когда (имена, а не ID);

CREATE PROCEDURE GetEmployeeClientOrders

AS

BEGIN

SELECT E.name AS employee\_name, C.name AS client\_name, O.order\_date

FROM Employees AS E

JOIN Delivery\_Points AS DP ON E.delivery\_point\_id = DP.delivery\_point\_id

JOIN Orders AS O ON DP.delivery\_point\_id = O.delivery\_point\_id

JOIN Clients AS C ON O.client\_id = C.client\_id;

END;

1. Понижение зарплаты на 20% всем сотрудникам ПВЗ с рейтингом меньше 3.5;

CREATE PROCEDURE UpdateEmployeeSalaryBasedOnDeliveryPointRating

AS

BEGIN

UPDATE Employees

SET salary = salary \* 0.8

WHERE delivery\_point\_id IN (

SELECT delivery\_point\_id

FROM Delivery\_Points

WHERE rating < 3.5

);

END;

1. Удаление всех ПВЗ с оборотом заказов меньше 1 в месяц со всеми их сотрудниками.

CREATE PROCEDURE DeleteDeliveryPointsWithEmployees

AS

BEGIN

WITH PointsToDelete AS (

SELECT dp.delivery\_point\_id

FROM Delivery\_Points dp

LEFT JOIN (

SELECT delivery\_point\_id, COUNT(\*) as monthly\_order\_count

FROM Orders

WHERE order\_date >= DATEADD(MONTH, DATEDIFF(MONTH, 0, GETDATE()), 0)

AND order\_date < DATEADD(MONTH, DATEDIFF(MONTH, -1, GETDATE()), 0)

GROUP BY delivery\_point\_id

) order\_counts ON dp.delivery\_point\_id = order\_counts.delivery\_point\_id

WHERE order\_counts.monthly\_order\_count < 1

)

DELETE FROM Employees

WHERE delivery\_point\_id IN (SELECT delivery\_point\_id FROM PointsToDelete);

DELETE FROM Delivery\_Points

WHERE delivery\_point\_id IN (SELECT delivery\_point\_id FROM PointsToDelete);

END;