***Часть 1***

***1. Выборка из одной Результат.***

* 1. Выбрать из произвольной Результат данные и отсортировать их по двум  произвольным имеющимся в таблице признакам (разные направления сортировки).

Запрос:

SELECT

id,

sum\_issued,

commission,

contract\_date,

redemption\_status

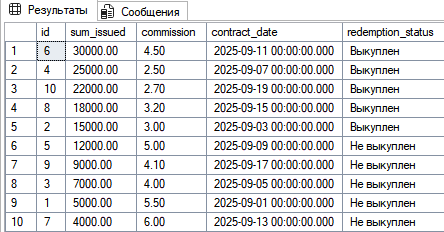
FROM Contract

ORDER BY

sum\_issued DESC,

contract\_date ASC;

Таблица:



1.2 Выбрать из произвольной Результат те записи, которые удовлетворяют

условию отбора (where). Привести 2-3 запроса.

Запросы:

SELECT

id,

sum\_issued,

redemption\_status,

client\_id,

manager\_id

FROM Contract

WHERE redemption\_status = 'Не выкуплен';

SELECT

id,

valuation,

depreciation,

product\_type\_id,

contract\_id

FROM Product

WHERE depreciation > 15;

SELECT

id,

price,

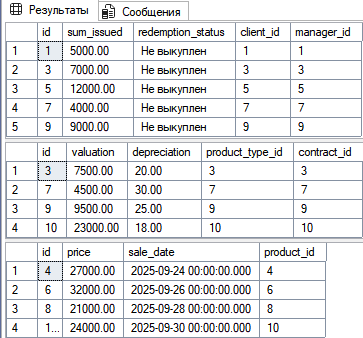
sale\_date,

product\_id

FROM Sale

WHERE price > 20000;

Результат:



1.3 Привести примеры 2-3 запросов с использованием агрегатных функций

(count, max, sum и др.) с группировкой и без группировки.

Запросы:

SELECT

SUM(sum\_issued) AS total\_sum\_issued

FROM Contract;

SELECT

MAX(price) AS max\_sale\_price

FROM Sale;

SELECT

sale\_date,

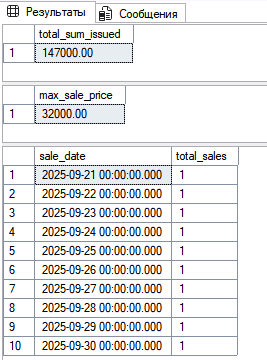
COUNT(\*) AS total\_sales

FROM Sale

GROUP BY sale\_date

ORDER BY sale\_date;

Результат:



1.4  Привести примеры подведения подытога с использованием GROUP BY [ALL] [ CUBE | ROLLUP](2-3 запроса). В ROLLUP и CUBE использовать не менее 2-х столбцов.

Запросы:

SELECT

manager\_id,

SUM(sum\_issued) AS total\_issued

FROM Contract

GROUP BY manager\_id

ORDER BY manager\_id;

SELECT

manager\_id,

redemption\_status,

SUM(sum\_issued) AS total\_issued

FROM Contract

GROUP BY ROLLUP (manager\_id, redemption\_status)

ORDER BY manager\_id, redemption\_status;

SELECT

manager\_id,

redemption\_status,

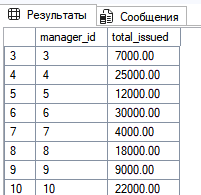
SUM(sum\_issued) AS total\_issued

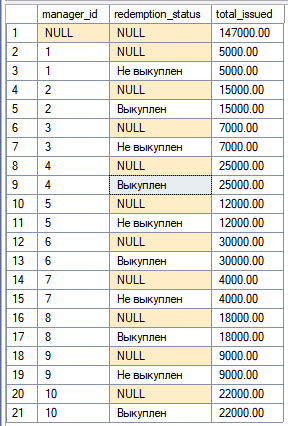
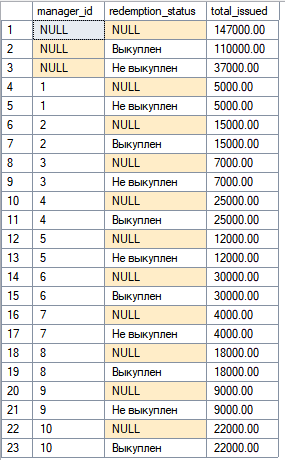
FROM Contract

GROUP BY CUBE (manager\_id, redemption\_status)

ORDER BY manager\_id, redemption\_status;

Результат:



1.5 Выбрать из таблиц информацию об объектах, в названиях которых нет заданной последовательности букв (LIKE).

Запросы:

SELECT

id,

name

FROM ProductType

WHERE name NOT LIKE N'%но%';

SELECT

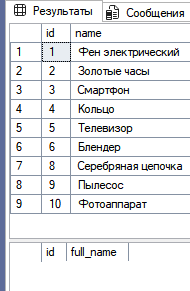
id,

full\_name

FROM Client

WHERE full\_name NOT LIKE N'%ов%';

Результат:



***2. Выборка из нескольких таблиц.***

2.1 Вывести информацию подчиненной (дочерней) Результат, заменяя коды

(значения внешних ключей) соответствующими символьными значениями из

родительских таблиц. Привести 2-3 запроса с использованием классического

подхода соединения таблиц (where).

Запросы:

SELECT

Contract.id AS ContractID,

Client.full\_name AS ClientName,

Manager.full\_name AS ManagerName,

Contract.sum\_issued,

Contract.redemption\_status

FROM Contract, Client, Manager

WHERE Contract.client\_id = Client.id

AND Contract.manager\_id = Manager.id;

SELECT

Product.id AS ProductID,

ProductType.name AS ProductType,

Material.name AS MaterialName,

ProductMaterial.weight

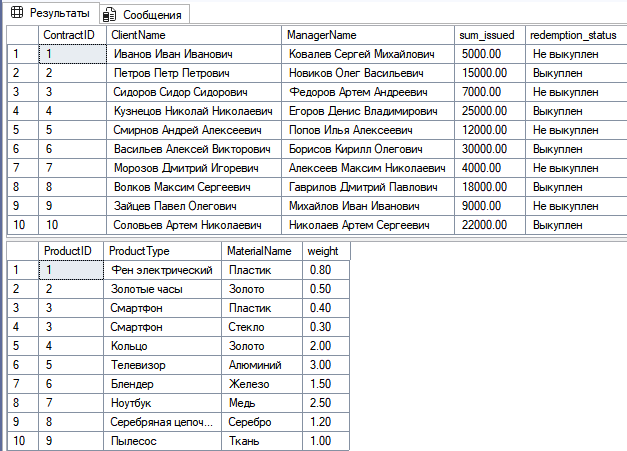
FROM ProductMaterial, Product, ProductType, Material

WHERE ProductMaterial.product\_id = Product.id

AND Product.product\_type\_id = ProductType.id

AND ProductMaterial.material\_id = Material.id;

Результат:



2.2. Реализовать запросы пункта 2.1 через внутреннее соединение inner join.

Запросы:

SELECT

c.id AS ContractID,

cl.full\_name AS ClientName,

m.full\_name AS ManagerName,

c.sum\_issued,

c.redemption\_status

FROM Contract c

INNER JOIN Client cl ON c.client\_id = cl.id

INNER JOIN Manager m ON c.manager\_id = m.id;

SELECT

p.id AS ProductID,

pt.name AS ProductType,

m.name AS MaterialName,

pm.weight

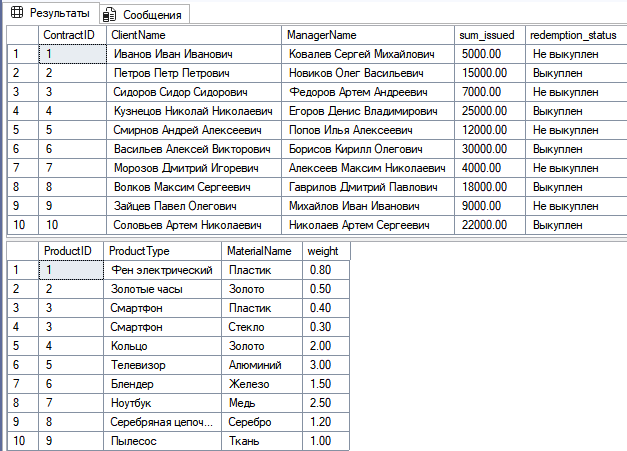
FROM ProductMaterial pm

INNER JOIN Product p ON pm.product\_id = p.id

INNER JOIN ProductType pt ON p.product\_type\_id = pt.id

INNER JOIN Material m ON pm.material\_id = m.id;

Результат:



2.3. Левое внешнее соединение left join. Привести 2-3 запроса.

Запросы:

SELECT

c.id AS ContractID,

cl.full\_name AS ClientName,

m.full\_name AS ManagerName,

c.sum\_issued,

c.redemption\_status

FROM Contract c

LEFT JOIN Client cl ON c.client\_id = cl.id

LEFT JOIN Manager m ON c.manager\_id = m.id;

SELECT

p.id AS ProductID,

pt.name AS ProductType,

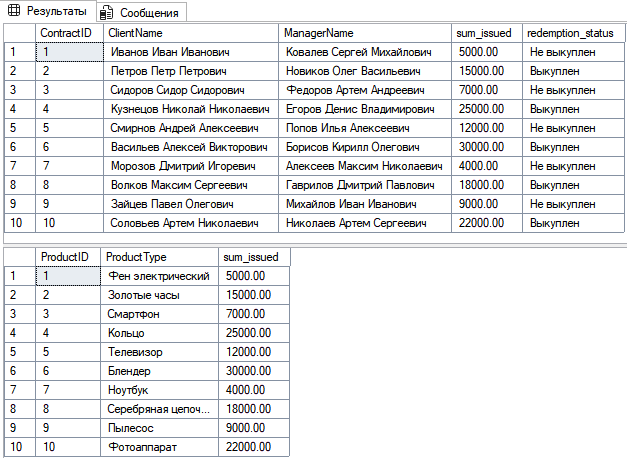
c.sum\_issued

FROM Product p

LEFT JOIN ProductType pt ON p.product\_type\_id = pt.id

LEFT JOIN Contract c ON p.contract\_id = c.id;

Результат:



2.4. Правое внешнее соединение right join. Привести 2-3 запроса

Запросы:

SELECT

c.id AS ContractID,

cl.full\_name AS ClientName,

m.full\_name AS ManagerName,

c.sum\_issued,

c.redemption\_status

FROM Contract c

RIGHT JOIN Client cl ON c.client\_id = cl.id

RIGHT JOIN Manager m ON c.manager\_id = m.id;

SELECT

p.id AS ProductID,

pt.name AS ProductType,

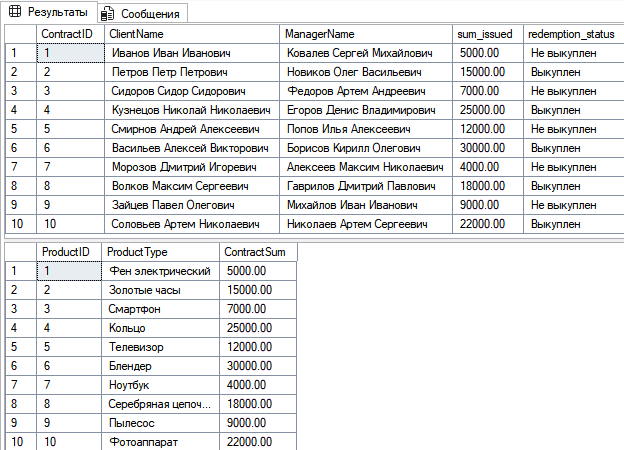
c.sum\_issued AS ContractSum

FROM Product p

RIGHT JOIN ProductType pt ON p.product\_type\_id = pt.id

RIGHT JOIN Contract c ON p.contract\_id = c.id;

Результат:



2.5. Привести примеры 2-3 запросов с использованием агрегатных функций

и группировки.

Запросы:

SELECT

manager\_id,

SUM(sum\_issued) AS TotalIssued

FROM Contract

GROUP BY manager\_id;

SELECT

redemption\_status,

COUNT(\*) AS NumContracts

FROM Contract

GROUP BY redemption\_status;

SELECT

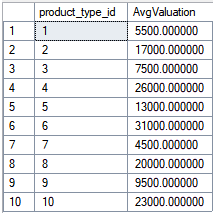
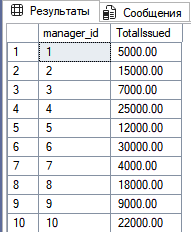
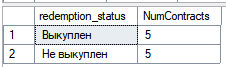
product\_type\_id,

AVG(valuation) AS AvgValuation

FROM Product

GROUP BY product\_type\_id;

Результат:

2.6. Привести примеры 2-3 запросов с использованием группировки и условия отбора групп (Having).

Запросы:

SELECT

manager\_id,

SUM(sum\_issued) AS TotalIssued

FROM Contract

GROUP BY manager\_id

HAVING SUM(sum\_issued) > 20000;

SELECT

material\_id,

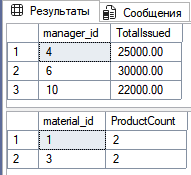
COUNT(product\_id) AS ProductCount

FROM ProductMaterial

GROUP BY material\_id

HAVING COUNT(product\_id) > 1;

Результат:



2.7. Привести примеры 3-4 вложенных (соотнесенных, c использованием IN, EXISTS) запросов.

Запросы:

SELECT full\_name, address

FROM Client

WHERE id IN (

SELECT client\_id

FROM Contract

WHERE sum\_issued > 20000

);

SELECT id, valuation

FROM Product

WHERE product\_type\_id IN (

SELECT product\_type\_id

FROM Product p

INNER JOIN Contract c ON p.contract\_id = c.id

WHERE c.sum\_issued > 15000

);

SELECT full\_name, passport\_data

FROM Client c

WHERE EXISTS (

SELECT 1

FROM Contract co

WHERE co.client\_id = c.id

AND co.redemption\_status = 'Выкуплен'

);

SELECT id, valuation

FROM Product p

WHERE EXISTS (

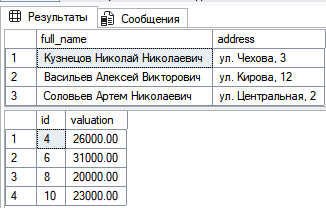
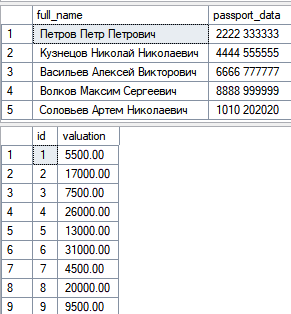
SELECT 1

FROM ProductMaterial pm

WHERE pm.product\_id = p.id

);

Результат:

***3. Представления***

3.1  На основе любых запросов из п. 2 создать два представления (VIEW).

Запросы:

CREATE VIEW vw\_ManagerTotalIssued AS

SELECT

manager\_id,

SUM(sum\_issued) AS TotalIssued

FROM Contract

GROUP BY manager\_id;

CREATE VIEW vw\_ContractDetails AS

SELECT

c.id AS ContractID,

cl.full\_name AS ClientName,

m.full\_name AS ManagerName,

c.sum\_issued,

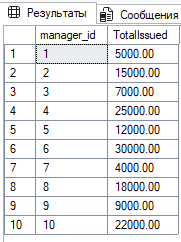
c.redemption\_status

FROM Contract c

RIGHT JOIN Client cl ON c.client\_id = cl.id

RIGHT JOIN Manager m ON c.manager\_id = m.id;

Результат:



3.2  Привести примеры использования общетабличных выражений (СТЕ) (2-3 запроса)

Запросы:

WITH ManagerTotals AS (

SELECT

manager\_id,

SUM(sum\_issued) AS TotalIssued

FROM Contract

GROUP BY manager\_id

)

SELECT \*

FROM ManagerTotals

WHERE TotalIssued > 20000;

WITH ClientContracts AS (

SELECT

client\_id,

COUNT(\*) AS NumContracts

FROM Contract

GROUP BY client\_id

)

SELECT cl.full\_name, cc.NumContracts

FROM ClientContracts cc

INNER JOIN Client cl ON cc.client\_id = cl.id

WHERE cc.NumContracts > 1;

WITH ProductMaterials AS (

SELECT

product\_id,

SUM(weight) AS TotalWeight

FROM ProductMaterial

GROUP BY product\_id

)

SELECT p.id AS ProductID, pt.name AS ProductType, pm.TotalWeight

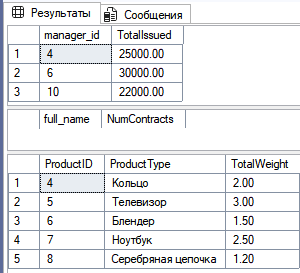
FROM ProductMaterials pm

INNER JOIN Product p ON pm.product\_id = p.id

INNER JOIN ProductType pt ON p.product\_type\_id = pt.id

WHERE pm.TotalWeight > 1.0;

Результат:



***4. Функции ранжирования***

4.1 Привести примеры 3-4 запросов с использованием ROW\_NUMBER, RANK, DENSE\_RANK (c  PARTITION BY и без)

Запросы:

SELECT

id AS ContractID,

client\_id,

sum\_issued,

ROW\_NUMBER() OVER (ORDER BY sum\_issued DESC) AS RowNum

FROM Contract;

SELECT

manager\_id,

client\_id,

sum\_issued,

RANK() OVER (PARTITION BY manager\_id ORDER BY sum\_issued DESC) AS RankPerManager

FROM Contract;

SELECT

id AS ProductID,

valuation,

DENSE\_RANK() OVER (ORDER BY valuation DESC) AS DenseRank

FROM Product;

SELECT

id AS ProductID,

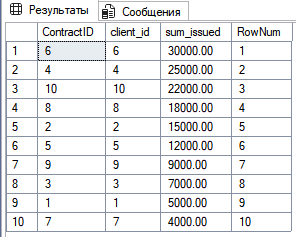
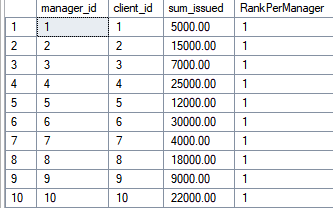
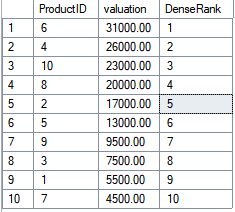
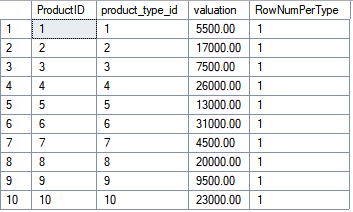
product\_type\_id,

valuation,

ROW\_NUMBER() OVER (PARTITION BY product\_type\_id ORDER BY valuation DESC) AS RowNumPerType

FROM Product;

Результат:

***5. Объдинение, пересечение, разность***

5.1 Привести примеры 3-4 запросов с использованием UNION / UNION ALL, EXCEPT, INTERSECT. Данные  в одном из запросов отсортируйте по

Запросы:

SELECT full\_name, passport\_data

FROM Client

UNION

SELECT full\_name, passport\_data

FROM Manager

ORDER BY full\_name;

SELECT full\_name, passport\_data

FROM Client

UNION ALL

SELECT full\_name, passport\_data

FROM Manager;

SELECT id AS ProductID

FROM Product

EXCEPT

SELECT product\_id

FROM ProductMaterial;

SELECT id AS ProductID

FROM Product

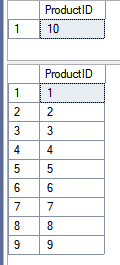
INTERSECT

SELECT product\_id

FROM ProductMaterial;

Результат:



***6. Использование CASE, PIVOT и UNPIVOT.***

6.1 Привести примеры получения сводных (итоговых) таблиц с использованием CASE

Запросы:

SELECT

manager\_id,

SUM(CASE WHEN redemption\_status = 'Выкуплен' THEN sum\_issued ELSE 0 END) AS TotalRedeemed,

SUM(CASE WHEN redemption\_status = 'Не выкуплен' THEN sum\_issued ELSE 0 END) AS TotalNotRedeemed,

SUM(sum\_issued) AS TotalIssued

FROM Contract

GROUP BY manager\_id;

SELECT

id AS ProductID,

valuation,

CASE

WHEN valuation < 10000 THEN 'Недорогой'

WHEN valuation BETWEEN 10000 AND 20000 THEN 'Средний'

ELSE 'Дорогой'

END AS PriceCategory

FROM Product;

SELECT

cl.id AS ClientID,

cl.full\_name,

COUNT(c.id) AS NumContracts,

CASE

WHEN COUNT(c.id) = 0 THEN 'Нет договоров'

WHEN COUNT(c.id) = 1 THEN 'Один договор'

ELSE 'Несколько договоров'

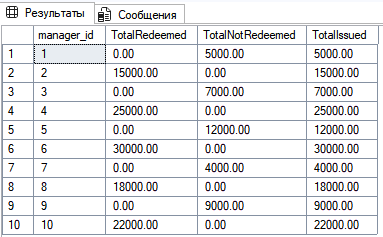
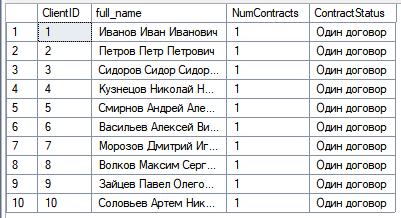
END AS ContractStatus

FROM Client cl

LEFT JOIN Contract c ON cl.id = c.client\_id

GROUP BY cl.id, cl.full\_name;

Результат:

6.2 Привести примеры получения сводных (итоговых) таблиц с использованием PIVOT и UNPIVOT.

Запросы:

SELECT \*

FROM (

SELECT manager\_id, sum\_issued, redemption\_status

FROM Contract

) AS SourceTable

PIVOT (

SUM(sum\_issued)

FOR redemption\_status IN ([Выкуплен], [Не выкуплен])

) AS PivotTable;

SELECT client\_id, Status, NumContracts

FROM (

SELECT client\_id, [Выкуплен], [Не выкуплен]

FROM (

SELECT client\_id, redemption\_status

FROM Contract

) AS SourceTable

PIVOT (

COUNT(redemption\_status)

FOR redemption\_status IN ([Выкуплен], [Не выкуплен])

) AS PivotTable

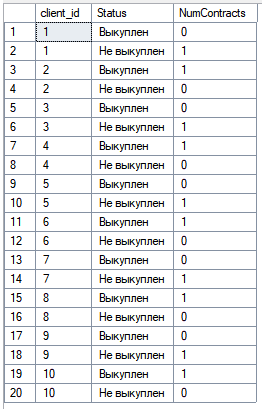
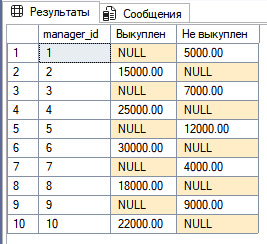
) AS p

UNPIVOT (

NumContracts FOR Status IN ([Выкуплен], [Не выкуплен])

) AS Unpvt;

Результат:

**Часть 2**

**Составить следующие запросы:**

a)  Выдать список товаров, выставленных на продажу

Запрос:

SELECT

p.id AS ProductID,

pt.name AS ProductType,

p.valuation AS Valuation,

p.depreciation AS Depreciation,

s.price AS SalePrice,

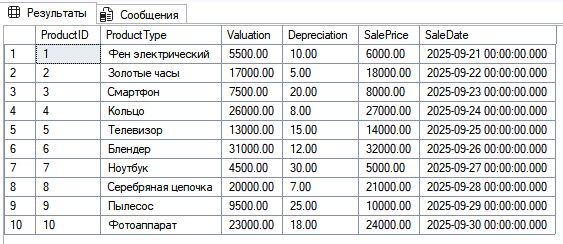
s.sale\_date AS SaleDate

FROM Sale s

JOIN Product p ON s.product\_id = p.id

JOIN ProductType pt ON p.product\_type\_id = pt.id;

Результат:



b)  Выдать список товаров, принятых в залог (дата, вид товара, количество)

Запрос:

SELECT

c.contract\_date AS AcceptanceDate,

pt.name AS ProductType,

COUNT(p.id) AS Quantity

FROM Product p

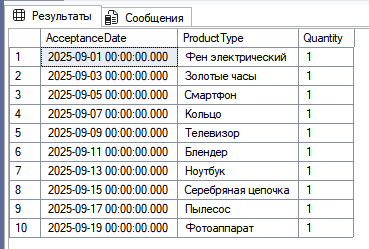
JOIN ProductType pt ON p.product\_type\_id = pt.id

JOIN Contract c ON p.contract\_id = c.id

GROUP BY c.contract\_date, pt.name

ORDER BY c.contract\_date;

Результат:



c)  Найти выручку ломбарда от комиссионных с начала текущего года для каждого вила товара

Запрос:  
SELECT

pt.name AS Вид\_товара,

SUM(c.sum\_issued \* c.commission / 100) AS Выручка\_от\_комиссионных

FROM Contract c

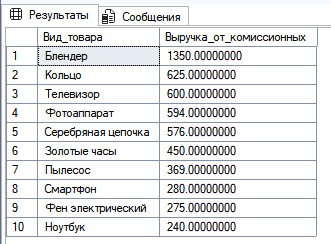
JOIN Product p ON c.id = p.contract\_id

JOIN ProductType pt ON p.product\_type\_id = pt.id

WHERE YEAR(c.contract\_date) = YEAR(GETDATE())

GROUP BY pt.name

ORDER BY Выручка\_от\_комиссионных DESC;

Результат:  


d)  Найти клиентов, которые не выкупили свой товар в срок

Запрос:

SELECT

cl.full\_name AS Клиент,

c.contract\_date AS Дата\_договора,

c.redemption\_term AS Срок\_выкупа,

c.redemption\_date AS Дата\_выкупа,

c.redemption\_status AS Статус

FROM Contract c

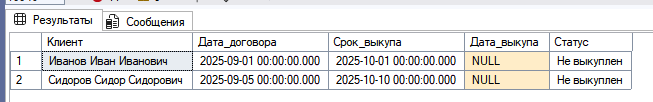
JOIN Client cl ON c.client\_id = cl.id

WHERE

(c.redemption\_status = 'Не выкуплен' AND c.redemption\_term < GETDATE())

OR (c.redemption\_date IS NOT NULL AND c.redemption\_date > c.redemption\_term);

Результат:



e)  Найти клиентов, пользовавшихся услугами ломбарда 2 и более раз и всегда выкупавших все свои товары

Запрос:

SELECT

cl.full\_name AS Клиент,

COUNT(c.id) AS Количество\_договоров

FROM Contract c

JOIN Client cl ON c.client\_id = cl.id

GROUP BY cl.full\_name

HAVING

COUNT(c.id) >= 2

AND SUM(CASE WHEN c.redemption\_status <> 'Выкуплен' THEN 1 ELSE 0 END) = 0;

Результат(была добавлена ещё одна запись, чтобы результат был не пустой):

