Оглавление

[Входные данные для всех запросов 3](#_Toc211697499)

[ЧАСТЬ 1 4](#_Toc211697500)

[1. Выборка из одной таблицы. 4](#_Toc211697501)

[1.1 Выбрать из произвольной таблицы данные и отсортировать их по двум произвольным имеющимся в таблице признакам (разные направления сортировки). 4](#_Toc211697502)

[1.2 Выбрать из произвольной таблицы те записи, которые удовлетворяют условию отбора (where). Привести 2-3 запроса. 5](#_Toc211697503)

[1.3 Привести примеры 2-3 запросов с использованием агрегатных функций (count, max, sum и др.) с группировкой и без группировки. 7](#_Toc211697504)

[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 8](#_Toc211697505)

[1.5 Выбрать из таблиц информацию об объектах, в названиях которых нет заданной последовательности букв (LIKE). 10](#_Toc211697506)

[2. Выборка из нескольких таблиц. 11](#_Toc211697507)

[2.1 Вывести информацию подчиненной (дочерней) таблицы, заменяя коды (значения внешних ключей) соответствующими символьными значениями из родительских таблиц. Привести 2-3 запроса с использованием классического подхода соединения таблиц where). 11](#_Toc211697508)

[2.2. Реализовать запросы пункта 2.1 через внутреннее соединение inner join. 13](#_Toc211697509)

[2.3. Левое внешнее соединение left join. Привести 2-3 запроса. 15](#_Toc211697510)

[2.4. Правое внешнее соединение right join. Привести 2-3 запроса 17](#_Toc211697511)

[2.5. Привести примеры 2-3 запросов с использованием агрегатных функций и группировки. 18](#_Toc211697512)

[2.6. Привести примеры 2-3 запросов с использованием группировки и условия отбора групп (Having). 21](#_Toc211697513)

[2.7. Привести примеры 3-4 вложенных (соотнесенных, c использованием IN, EXISTS) запросов. 22](#_Toc211697514)

[3. Представления 25](#_Toc211697515)

[3.1 На основе любых запросов из п. 2 создать два представления (VIEW). 25](#_Toc211697516)

[3.2 Привести примеры использования общетабличных выражений (СТЕ) (2-3 запроса) 26](#_Toc211697517)

[4. Функции ранжирования 28](#_Toc211697518)

[4.1 Привести примеры 3-4 запросов с использованием ROW\_NUMBER, RANK, DENSE\_RANK (c  PARTITION BY и без) 28](#_Toc211697519)

[5. Объдинение, пересечение, разность 30](#_Toc211697520)

[5.1 Привести примеры 3-4 запросов с использованием UNION / UNION ALL, EXCEPT, INTERSECT. Данные в одном из запросов отсортируйте по произвольному признаку. 30](#_Toc211697521)

[6. Использование CASE, PIVOT и UNPIVOT. 32](#_Toc211697522)

[6.1 Привести примеры получения сводных (итоговых) таблиц с использованием CASE 32](#_Toc211697523)

[Классификация клиентов по общей сумме кредитов 32](#_Toc211697524)

[6.2 Привести примеры получения сводных (итоговых) таблиц с использованием PIVOT и UNPIVOT. 34](#_Toc211697525)

[ЧАСТЬ 2 37](#_Toc211697526)

[a) Для каждого клиента вывести количество взятых им кредитов и их общую сумму 37](#_Toc211697527)

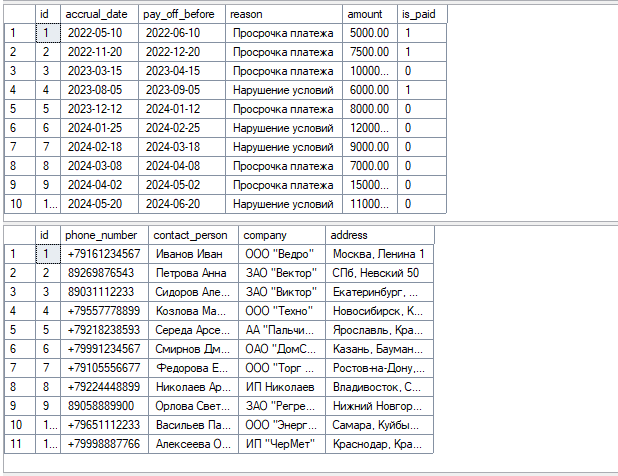
[b) Самые выгодные кредиты 37](#_Toc211697528)

[c) Количество сделок по дням текущего месяца 38](#_Toc211697529)

[d) Непопулярные виды кредитов 38](#_Toc211697530)

[e) Клиенты с наибольшим количеством кредитов и плохой историей 39](#_Toc211697531)

# Входные данные для всех запросов



# ЧАСТЬ 1

## 1. Выборка из одной таблицы.

### 1.1 Выбрать из произвольной таблицы данные и отсортировать их по двум произвольным имеющимся в таблице признакам (разные направления сортировки).

Все клиенты банка упорядоченные по имени контактного человека по алфавиту

SELECT

\*

FROM

Client

ORDER BY

contact\_person



Все клиенты банка упорядоченные по компании в обратном порядке

SELECT

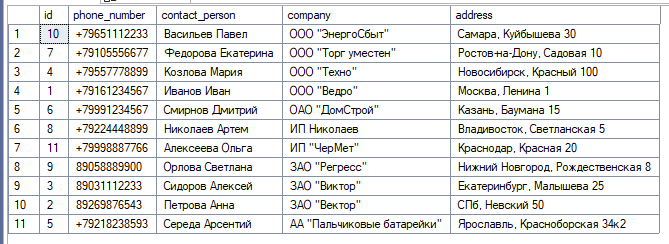
\*

FROM

Client

ORDER BY

company DESC



### 1.2 Выбрать из произвольной таблицы те записи, которые удовлетворяют условию отбора (where). Привести 2-3 запроса.

Все записи кредитной истории наших клиентов, которые сотрудничали со Сбербанком

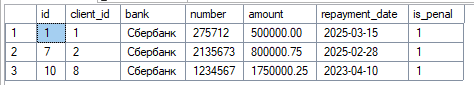
SELECT

\*

FROM

[Credit history]

WHERE

 bank = 'Сбербанк'

Все кредитные продукты минимальный оъбем которых 50000, а максимальный срок не более 50 месяцев

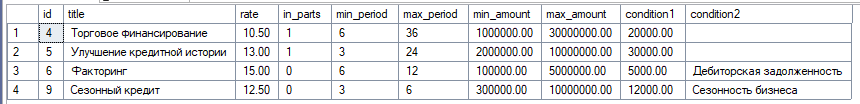
SELECT

\*

FROM

[Credit product]

WHERE

 min\_amount > 50000 AND max\_period < 50

Штрафы наложенные за 2024 год в размере больше 7000

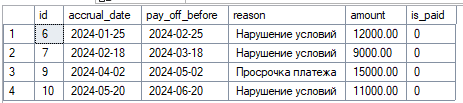
SELECT

\*

FROM

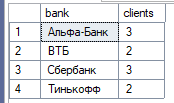
Penalty

WHERE

 (accrual\_date BETWEEN '2024-01-01' AND '2024-12-31') AND amount > 7000

### 1.3 Привести примеры 2-3 запросов с использованием агрегатных функций (count, max, sum и др.) с группировкой и без группировки.

Сколько записей в кредитной истории принадежит каждому из банков

SELECT

bank,

COUNT(\*) as clients

FROM

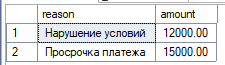
[Credit history]

GROUP BY

bank

Наибольший размер штрафа для каждой из причин его получения

SELECT

 reason,

MAX(amount) as amount

FROM

Penalty

GROUP BY

reason

Общая сумма кредитов взятых в банке

SELECT

 SUM(amount) as amount

FROM

Deal

### 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

Статистика штрафов с агрегацией по годам и типам нарушений

SELECT

CASE

WHEN GROUPING(YEAR(accrual\_date)) = 1 THEN 'Все года'

ELSE CAST(YEAR(accrual\_date) AS NVARCHAR(10))

END AS year,

CASE

WHEN GROUPING(reason) = 1 THEN 'Все причины'

ELSE reason

END AS reason,

SUM(amount) AS total\_amount

FROM

Penalty

GROUP BY

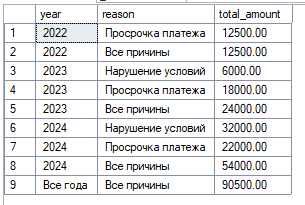
ROLLUP(YEAR(accrual\_date), reason)

ORDER BY

GROUPING(YEAR(accrual\_date)),

YEAR(accrual\_date),

GROUPING(reason),

reason;

Сводка кредитной активности по всем срезам: банки, годы, платежная дисциплина

SELECT

CASE

WHEN GROUPING(bank) = 1 THEN 'Все банки'

ELSE bank

END AS bank,

CASE

WHEN GROUPING(YEAR(repayment\_date)) = 1 THEN 'Все года'

ELSE CAST(YEAR(repayment\_date) AS NVARCHAR(10))

END AS repayment\_date,

CASE

WHEN GROUPING(is\_penal) = 1 THEN 'Все уплаты/неуплаты'

ELSE CAST(is\_penal AS NVARCHAR(2))

END AS is\_penal,

SUM(amount) as amount

FROM

[Credit history]

GROUP BY

CUBE(bank, YEAR(repayment\_date), is\_penal)

ORDER BY

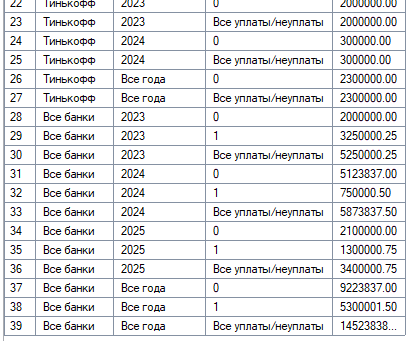
GROUPING(bank),

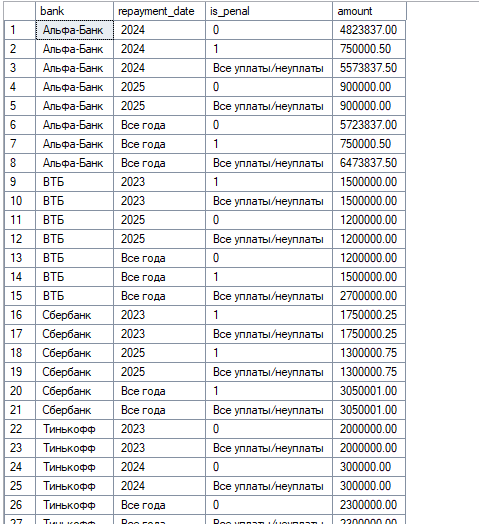
bank,

GROUPING(YEAR(repayment\_date)),

repayment\_date,

GROUPING(is\_penal),

 is\_penal;



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

Клиенты фамилии контактных лиц которых заканчиваются не на "ров", "рова", "ев" или "ева"

SELECT

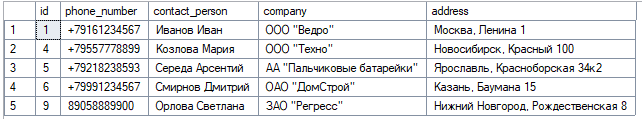
\*

FROM

Client

WHERE

contact\_person NOT LIKE('%ров %') AND contact\_person NOT LIKE ('%ев %') AND

 contact\_person NOT LIKE('%рова %') AND contact\_person NOT LIKE ('%ева %')

Клиенты названия компаний которых не являются "ЗАО" или "ИП"

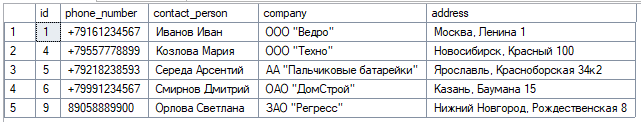
SELECT

\*

FROM

Client

WHERE

 company NOT LIKE('ЗАО "%') AND company NOT LIKE ('ИП "%')

## 2. Выборка из нескольких таблиц.

### 2.1 Вывести информацию подчиненной (дочерней) таблицы, заменяя коды (значения внешних ключей) соответствующими символьными значениями из родительских таблиц. Привести 2-3 запроса с использованием классического подхода соединения таблиц where).

Краткий вывод кредитной истории клиентов с именами кредиторов

SELECT

c.contact\_person, cr.bank, cr.amount

FROM

Client c, [Credit history] cr

WHERE

 c.id = cr.client\_id;

Вывод существующих сделок с указанием названия выбранног кредитного продукта

SELECT

d.amount, cp.title

FROM

Deal d, [Credit product] cp

WHERE

d.product\_id = cp.id;

Вывод размера сделок с поступившими на них платежами

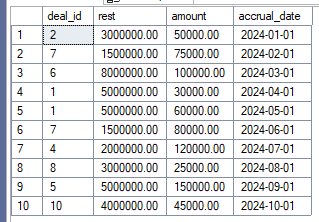
SELECT

p.deal\_id, d.rest, p.amount, p.accrual\_date

FROM

Payment p, Deal d

WHERE

 p.deal\_id = d.id;

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

Краткий вывод кредитной истории клиентов с названиями кредиторов

SELECT

c.company, cr.bank, cr.amount

FROM

 Client c INNER JOIN [Credit history] cr ON c.id=cr.client\_id

Вывод существующих сделок с указанием названия выбранного кредитного продукта

SELECT

d.amount, cp.title

FROM

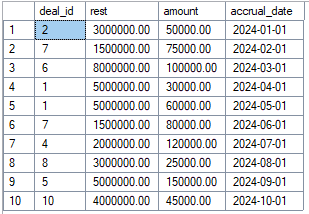
 Deal d INNER JOIN [Credit product] cp ON d.product\_id = cp.id;

Вывод размера сделок с поступившими на них платежами

SELECT

p.deal\_id, d.rest, p.amount, p.accrual\_date

FROM

 Payment p INNER JOIN Deal d ON p.deal\_id = d.id;

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

Краткий вывод кредитной истории клиентов с названиями кредиторов

SELECT

c.company,

cr.bank,

cr.amount

FROM

Client c LEFT JOIN [Credit history] cr

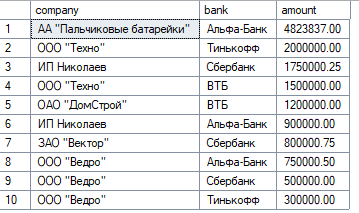
ON c.id = cr.client\_id

WHERE

cr.amount IS NOT NULL

ORDER BY

cr.amount DESC;

Все сделки и платежи по ним

SELECT

d.id as deal\_id,

d.amount as deal\_amount,

d.deal\_start,

pm.amount as payment\_amount,

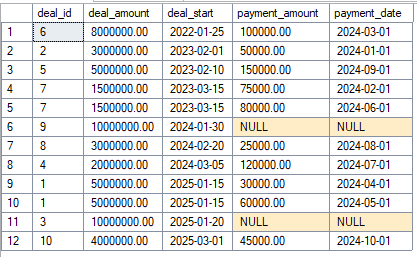
pm.accrual\_date as payment\_date

FROM

Deal d LEFT JOIN Payment pm

ON d.id = pm.deal\_id

ORDER BY

d.deal\_start ;

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

Записи в кредитной истории вида: Банк, клиент, долг

SELECT

ch.bank,

c.company,

ch.amount

FROM

Client c

RIGHT JOIN [Credit history] ch ON c.id = ch.client\_id;

Кредитные продукты и клиенты которые их взяли

SELECT

cpr.title,

c.contact\_person,

c.company

FROM

Client c RIGHT JOIN Client\_product cp

ON c.id = cp.client\_id

RIGHT JOIN [Credit product] cpr ON cp.product\_id = cpr.id;

 Все штрафы и соответствующие платежи

SELECT

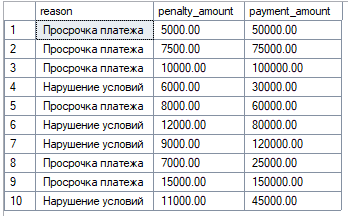
pn.reason,

pn.amount as penalty\_amount,

pm.amount as payment\_amount

FROM

Payment pm RIGHT JOIN Penalty pn

 ON pm.penalty\_id = pn.id;

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

Статистика использования кредитных продуктов

SELECT

cp.title,

COUNT(d.id) as deal\_count,

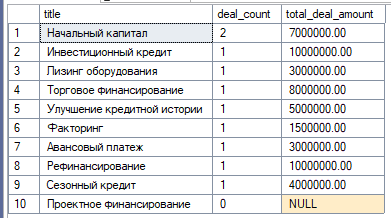
SUM(d.amount) as total\_deal\_amount

FROM

[Credit product] cp

LEFT JOIN Deal d ON cp.id = d.product\_id

GROUP BY

 cp.title, cp.id;

Анализ штрафов по клиентам

SELECT

c.contact\_person,

COUNT(pn.id) as penalty\_count,

SUM(pn.amount) as total\_penalties,

SUM(CASE WHEN pn.is\_paid = 1 THEN pn.amount ELSE 0 END) as paid\_penalties

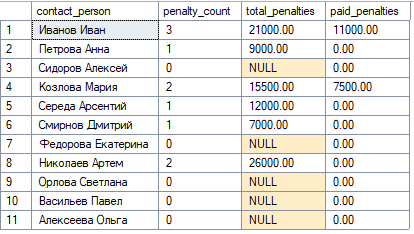
FROM

Client c

LEFT JOIN [Credit history] ch ON c.id = ch.client\_id

LEFT JOIN Penalty pn ON ch.id = pn.id

GROUP BY

 c.id, c.contact\_person;

Суммарная статистика кредитов по банкам

SELECT

c.company,

ch.bank,

SUM(amount) as total\_amount

FROM

[Credit history] ch LEFT JOIN Client c

ON ch.client\_id = c.id

GROUP BY

c.company,

 ch.bank;

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

Клиенты с общей суммой кредитов более 1 млн

SELECT

c.contact\_person,

SUM(ch.amount) as total\_credit

FROM

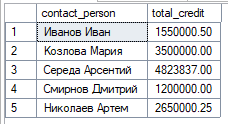
Client c JOIN [Credit history] ch

ON c.id = ch.client\_id

GROUP BY

c.id, c.contact\_person

HAVING

 SUM(ch.amount) > 1000000;

Банки из кредитной истории общая сумма взятых кредитов клиентами у которых больше 500000

SELECT

bank,

AVG(amount) as avg\_credit\_amount,

COUNT(\*) as credit\_count

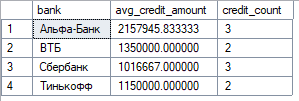
FROM

[Credit history]

GROUP BY

bank

HAVING

 AVG(amount) > 500000;

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

Сделки, которые больше средней по своему продукту

SELECT

d.id,

d.amount,

cp.title

FROM

Deal d

JOIN [Credit product] cp ON d.product\_id = cp.id

WHERE

EXISTS (

SELECT 1

FROM Deal avg\_deal

WHERE avg\_deal.product\_id = cp.id

HAVING d.amount > AVG(avg\_deal.amount)

);

Клиенты, у которых есть неоплаченные штрафы

SELECT

c.contact\_person,

c.company,

c.address

FROM

Client c

WHERE

EXISTS (

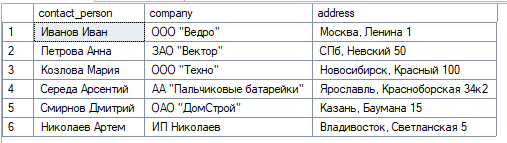
SELECT 1

FROM [Credit history] ch

JOIN Penalty p ON ch.id = p.id

WHERE ch.client\_id = c.id

AND p.is\_paid = 0

 );

Кредитные продукты, по которым не заключено ни одной сделки

SELECT

title

FROM

[Credit product]

WHERE

id NOT IN

(SELECT DISTINCT

 product\_id

FROM

Deal

WHERE

product\_id IS NOT NULL);

Клиенты, которые вносили платежи

SELECT

contact\_person

FROM

Client c

WHERE

EXISTS (

SELECT

1

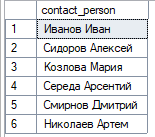
FROM

Deal d

JOIN Payment p ON d.id = p.deal\_id

WHERE

d.client\_id = c.id

 );

## 3. Представления

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

Кредитные продукты и клиенты которые их взяли

GO

CREATE VIEW ClientProductInterests AS

SELECT

cpr.title,

c.contact\_person,

c.company

FROM

Client c

RIGHT JOIN Client\_product cp ON c.id = cp.client\_id

RIGHT JOIN [Credit product] cpr ON cp.product\_id = cpr.id;

GO

SELECT \* FROM ClientProductInterests

Анализ штрафов по клиентам

GO

CREATE VIEW ClientPenaltyAnalysis AS

SELECT

c.contact\_person,

COUNT(pn.id) as penalty\_count,

SUM(pn.amount) as total\_penalties,

SUM(CASE WHEN pn.is\_paid = 1 THEN pn.amount ELSE 0 END) as paid\_penalties

FROM

Client c

LEFT JOIN [Credit history] ch ON c.id = ch.client\_id

LEFT JOIN Penalty pn ON ch.id = pn.id

GROUP BY

c.id, c.contact\_person;

GO

SELECT \* FROM ClientPenaltyAnalysis

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

Банки со средним кредитом более 500000

WITH HighAvgBanks AS (

SELECT

bank,

AVG(amount) as avg\_credit\_amount,

COUNT(\*) as credit\_count

FROM

[Credit history]

GROUP BY

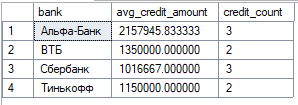
bank

HAVING

AVG(amount) > 500000

)

SELECT \*

FROM HighAvgBanks;

Записи в кредитной истории вида: Банк, клиент, долг

WITH BankClientDebts AS (

SELECT

ch.bank,

c.company,

ch.amount

FROM

Client c

RIGHT JOIN [Credit history] ch ON c.id = ch.client\_id

)

SELECT \* FROM BankClientDebts;



## 4. Функции ранжирования

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

Нумерация записей клиентов в крединой истории по банкам и величине взятых кредитов

SELECT

LEFT(ch.bank,15) AS bank,

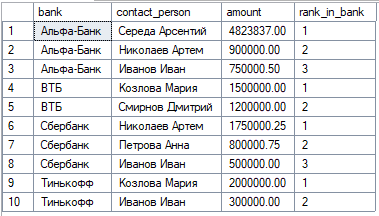
LEFT(c.contact\_person,20) AS contact\_person,

LEFT(ch.amount, 15) AS amount,

RANK() OVER (PARTITION BY ch.bank ORDER BY ch.amount DESC) as rank\_in\_bank

FROM

Client c

JOIN [Credit history] ch ON c.id = ch.client\_id;

Нумерация кредитных продуктов по процентной ставке внутри каждой категории (с рассрочкой/без рассрочки)

SELECT

LEFT(title,20) AS title,

LEFT(rate,10) AS rate,

DENSE\_RANK() OVER (PARTITION BY in\_parts ORDER BY rate DESC) as dense\_rank\_by\_parts

FROM [Credit product];

Нумерация сделок по времени кредитования и размеру кредита

SELECT

LEFT(c.contact\_person,20) AS contact\_person,

LEFT(d.amount,15) AS amount,

LEFT(d.period\_, 10) AS \_period,

ROW\_NUMBER() OVER (

PARTITION BY d.client\_id

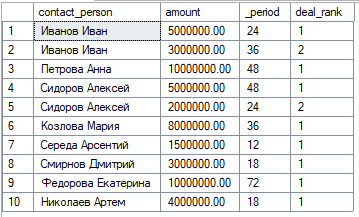
ORDER BY d.amount DESC, d.period\_ ASC

) as deal\_rank

FROM

Deal d

JOIN Client c ON d.client\_id = c.id;



## 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 произвольному признаку.

Клиенты которые заключали сделки или имеют записи в кредитной истории

SELECT

contact\_person FROM Client

WHERE

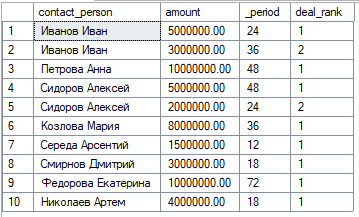
id IN (SELECT client\_id FROM [Credit history])

UNION

SELECT

contact\_person

FROM

 Client

WHERE

id IN (SELECT client\_id FROM Deal);

Объединение продуктов со временем кредитования не менее 24мес и возможностью оплаты по частям с дублями

SELECT

title, rate, in\_parts, min\_period, max\_period, min\_amount, max\_amount

FROM

[Credit product]

WHERE

in\_parts = 1

UNION ALL

SELECT

title, rate, in\_parts, min\_period, max\_period, min\_amount, max\_amount

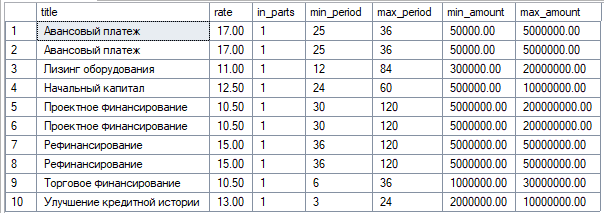
FROM

[Credit product]

WHERE

min\_period > 24

ORDER BY

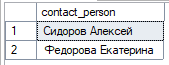
 title

Клиенты у которых есть активные сделки, но нет кредитной истории

SELECT

contact\_person FROM Client

WHERE

 id IN (SELECT client\_id FROM Deal WHERE rest > 0)

EXCEPT

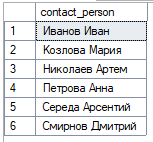
SELECT

contact\_person FROM Client

WHERE

id IN (SELECT client\_id FROM [Credit history]);

Клиенты которые и брали кредиты и заключали сделки

SELECT

contact\_person FROM Client

WHERE

id IN (SELECT client\_id FROM [Credit history])

INTERSECT

SELECT

contact\_person FROM Client

WHERE

id IN (SELECT client\_id FROM Deal);

# 6. Использование CASE, PIVOT и UNPIVOT.

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

## Классификация клиентов по общей сумме кредитов

SELECT

contact\_person,

company,

(SELECT SUM(amount) FROM Deal d WHERE client\_id = c.id) as total\_credit,

CASE

WHEN (SELECT SUM(amount) FROM Deal WHERE client\_id = c.id) > 8000000 THEN ‘Заслуженный кредитор'

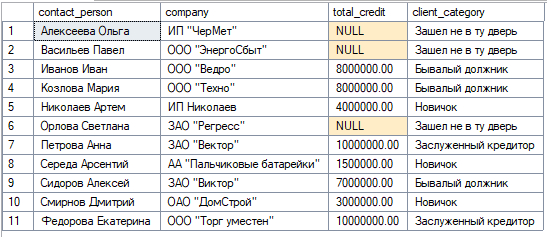
WHEN (SELECT SUM(amount) FROM Deal WHERE client\_id = c.id) > 5000000 THEN 'Бывалый должник'

WHEN (SELECT SUM(amount) FROM Deal WHERE client\_id = c.id) > 0 THEN 'Новичок'

ELSE 'Зашел не в ту дверь'

END as client\_category

FROM Client c

ORDER BY c.contact\_person;

Подсчет колличества сделок по категориям их продолжительнсти

SELECT

CASE

WHEN period\_ <= 12 THEN 'Краткосрочные (до 1 года)'

WHEN period\_ <= 36 THEN 'Среднесрочные (1-3 года)'

ELSE 'Долгосрочные (более 3 лет)'

END as period\_category,

COUNT(\*) as deal\_count,

AVG(amount) as avg\_amount,

SUM(amount) as total\_amount

FROM Deal

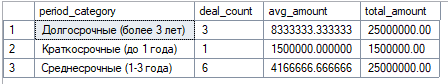
GROUP BY

CASE

WHEN period\_ <= 12 THEN 'Краткосрочные (до 1 года)'

WHEN period\_ <= 36 THEN 'Среднесрочные (1-3 года)'

ELSE 'Долгосрочные (более 3 лет)'

 END;

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

Количество клиентов по типам компаний

SELECT \*

FROM (

SELECT

contact\_person,

CASE

WHEN company LIKE 'ООО %' THEN 'ООО'

WHEN company LIKE 'АО %' THEN 'АО'

WHEN company LIKE 'ЗАО %' THEN 'АО'

ELSE 'Другие'

END as company\_type

 FROM Client

) AS SourceTable

PIVOT (

COUNT(contact\_person)

FOR company\_type IN ([ООО], [АО], [Другие])

) AS pvt;

Развернутые с UNPIVOT данные о кредитных продуктах

SELECT product\_id, parameter\_name, parameter\_value

FROM (

SELECT

id as product\_id,

CAST(min\_amount AS NVARCHAR(20)) as min\_amount,

CAST(max\_amount AS NVARCHAR(20)) as max\_amount,

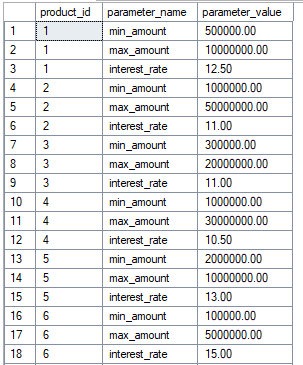
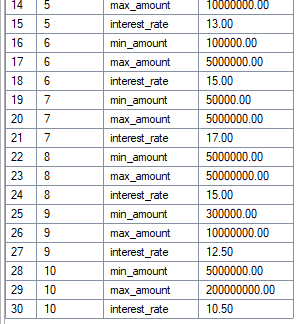
CAST(rate AS NVARCHAR(20)) as interest\_rate

FROM [Credit product]

) AS SourceTable

UNPIVOT (

parameter\_value FOR parameter\_name IN (min\_amount, max\_amount, interest\_rate)

) AS Upvt;

Развернутые с UNPIVOT данные о клиентах вида

SELECT client\_id, condition\_type, condition\_value

FROM (

SELECT

id as client\_id,

CAST(LEN(contact\_person) AS NVARCHAR(10)) as name\_length,

CAST(LEN(company) AS NVARCHAR(10)) as company\_name\_length,

CAST(

CASE

WHEN phone\_number LIKE '+7%' THEN '1'

ELSE '0'

END AS NVARCHAR(10)

) as phone\_format\_numeric

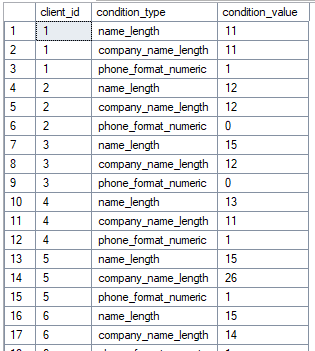
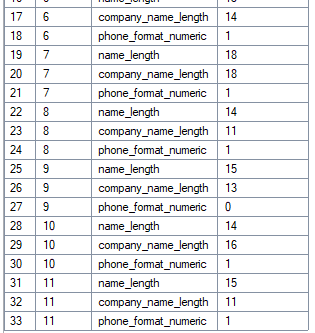
FROM Client

) AS SourceTable

UNPIVOT (

condition\_value FOR condition\_type IN (name\_length, company\_name\_length, phone\_format\_numeric)

) AS upvt;



# ЧАСТЬ 2

## a) Для каждого клиента вывести количество взятых им кредитов и их общую сумму

SELECT

c.contact\_person,

c.company,

COUNT(ch.id) as credit\_count,

SUM(ch.amount) as total\_amount

FROM

Client c

LEFT JOIN [Credit history] ch ON c.id = ch.client\_id

GROUP BY

 c.id, c.contact\_person, c.company;

## b) Самые выгодные кредиты

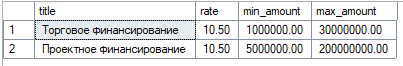
SELECT

title,

rate,

min\_amount,

max\_amount

FROM

[Credit product]

WHERE

in\_parts = 1

AND rate = (SELECT MIN(rate) FROM [Credit product] WHERE in\_parts = 1);

## c) Количество сделок по дням текущего месяца

SELECT

deal\_start,

COUNT(\*) as deal\_count

FROM

Deal

WHERE

deal\_start >= DATEFROMPARTS(YEAR(GETDATE()), MONTH(GETDATE()), 1)

GROUP BY

deal\_start

ORDER BY

deal\_start;

## d) Непопулярные виды кредитов

SELECT

title

FROM

[Credit product]

WHERE

id NOT IN

 (SELECT DISTINCT

product\_id

FROM

Deal

WHERE

product\_id IS NOT NULL);

## e) Клиенты с наибольшим количеством кредитов и плохой историей

SELECT TOP 1 W

c.contact\_person,

c.company,

COUNT(ch.id) as credit\_count,

SUM(CASE WHEN ch.is\_penal = 1 THEN 1 ELSE 0 END) as penal\_credits

FROM

Client c

JOIN [Credit history] ch ON c.id = ch.client\_id

GROUP BY

c.id, c.contact\_person, c.company

HAVING

COUNT(ch.id) = (

SELECT MAX(credit\_count)

FROM (

SELECT COUNT(id) as credit\_count

FROM [Credit history]

GROUP BY client\_id

) as counts

)

 AND SUM(CASE WHEN ch.is\_penal = 1 THEN 1 ELSE 0 END) > 0;