лаб 6

Клиенты, заключившие более 1-го договора с начала текущего года

```sql
-- Реляционная модель:
SELECT
    cl.Название AS Клиент,
    COUNT(c.ID) AS Количество_Договоров
FROM
    Client cl
    JOIN Contract c ON cl.ID = c.ID_Клиента
WHERE
    YEAR(c.Дата_Заключения) = YEAR(GETDATE())
GROUP BY
    cl.ID, cl.Название
HAVING
    COUNT(c.ID) > 1
ORDER BY
    Количество_Договоров DESC;
```

```sql
-- Графовая модель:
SELECT
    client.name AS Клиент,
    COUNT(DISTINCT contract.id) AS Количество_Договоров
FROM
    client_node AS client,
    contract_node AS contract
WHERE
    MATCH(client-(conclude)->contract)
    AND YEAR(contract.start_date) = YEAR(GETDATE())
GROUP BY
    client.id, client.name
HAVING
    COUNT(DISTINCT contract.id) > 1
ORDER BY
    Количество_Договоров DESC;
```

Клиенты, которые арендуют площади только на 1-х этажах

```sql
-- Реляционная модель:
SELECT cl.Название AS Клиент
FROM Client cl
    JOIN Contract c ON cl.ID = c.ID_Клиента
    JOIN RetailPoint rp ON c.ID_Точки = rp.ID
GROUP BY cl.ID, cl.Название
HAVING MIN(rp.Этаж) = 1
    AND MAX(rp.Этаж) = 1
ORDER BY cl.Название;
```

```sql
-- Графовая модель:
SELECT
    client.name AS Клиент
FROM
    client_node AS client,
    contract_node AS contract,
    retail_point_node AS retail_point
WHERE
    MATCH(client-(conclude)->contract<-(rented)-retail_point)
GROUP BY
    client.id, client.name
HAVING
    MIN(retail_point.floor) = 1
    AND MAX(retail_point.floor) = 1
ORDER BY
    client.name;
```

Клиенты, имеющие задолженности по уплате аренды более 1 месяца

```sql
 -- Реляционная модель:
SELECT
    cl.Название AS Клиент,
    COUNT(p.ID) AS Количество_Просроченных_Платежей,
    SUM(p.Сумма) AS Сумма_Задолженности
FROM
    Client cl
    JOIN Contract c ON cl.ID = c.ID_Клиента
    JOIN Payment p ON c.ID = p.ID_Договора
WHERE
    p.Статус = 0
GROUP BY
    cl.ID, cl.Название
HAVING
    COUNT(DISTINCT p.Месяц) > 1
```

```sql
-- Графовая модель:
SELECT
    client.name AS Клиент,
    COUNT(DISTINCT payment.id) AS Количество_Просроченных_Платежей,
    SUM(payment.amount) AS Сумма_Задолженности
FROM
    client_node AS client,
    contract_node AS contract,
    payment_node AS payment
WHERE
    MATCH(client-(conclude)->contract-(defines)->payment)
      AND payment.status = 0
GROUP BY
    client.id, client.name
HAVING
    COUNT(DISTINCT payment.month) > 1
```

Торговые точки, на которые не было заключено ни одного договора в течение последнего года

```sql
-- Реляционная модель:
SELECT
    rp.Адрес,
    rp.Этаж,
    rp.Площадь,
    rp.Стоимость_Аренды
FROM
    RetailPoint rp
WHERE
    rp.ID NOT IN (
        SELECT DISTINCT c.ID_Точки
        FROM Contract c
        WHERE c.Дата_Заключения >= DATEADD(YEAR, -1, GETDATE())
    )
    AND rp.Статус = 1  -- Свободные точки
ORDER BY
    rp.Этаж, rp.Стоимость_Аренды DESC;
```

```sql
-- Графовая модель:
SELECT
    point.address AS Адрес,
    point.floor AS Этаж,
    point.area AS Площадь,
    point.rental_cost AS Стоимость_Аренды
FROM
    retail_point_node AS point
WHERE
    point.status = 1
    AND NOT EXISTS (
        SELECT 1
        FROM
            retail_point_node AS rp,
            contract_node AS con
        WHERE
            MATCH(rp-(rented)->con)
            AND rp.id = point.id
            AND con.start_date >= DATEADD(YEAR, -1, GETDATE())
    )
ORDER BY
    point.floor, point.rental_cost DESC;
```

Клиенты, заключившие наибольшее количество договоров на аренду

```sql
-- Реляционная модель:
SELECT
    cl.Название AS Клиент,
    COUNT(c.ID) AS Количество_Договоров,
    SUM(c.Финальная_Стоимость) AS Общая_Стоимость
FROM
    Client cl
    JOIN Contract c ON cl.ID = c.ID_Клиента
GROUP BY
    cl.ID, cl.Название
HAVING
    COUNT(c.ID) = (
        SELECT MAX(Договоры)
        FROM (
            SELECT COUNT(c2.ID) AS Договоры
            FROM Client cl2
            JOIN Contract c2 ON cl2.ID = c2.ID_Клиента
            GROUP BY cl2.ID
        ) AS MaxCount
    )
ORDER BY
    Общая_Стоимость DESC;
```

```sql
-- Графовая модель:
WITH client_contracts AS (
    SELECT
        client.id,
        client.name,
        COUNT(DISTINCT contract.id) AS contract_count,
        SUM(contract.final_cost) AS total_cost
    FROM
        client_node AS client,
        contract_node AS contract
    WHERE
        MATCH(client-(conclude)->contract)
    GROUP BY
        client.id, client.name
),
max_contract_count AS (
    SELECT MAX(contract_count) AS max_count
    FROM client_contracts
)
SELECT
    cc.name AS Клиент,
    cc.contract_count AS Количество_Договоров,
    cc.total_cost AS Общая_Стоимость
FROM
    client_contracts cc
    CROSS JOIN max_contract_count mcc
WHERE
    cc.contract_count = mcc.max_count
ORDER BY
    cc.total_cost DESC;
```