**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА   
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ**

**ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**НИЖЕГОРОДСКИЙ ИНСТИТУТ УПРАВЛЕНИЯ – ФИЛИАЛ**

ОТЧЕТ

По направлению: Многомерный анализ данных

Выполнил студент группы Ик-731

Соколов Д.А

г. Нижний Новгород

2025

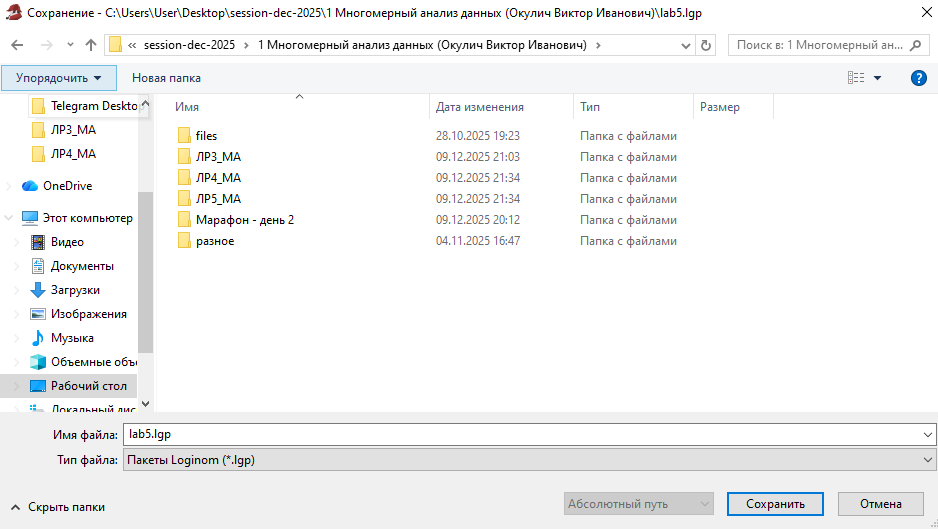
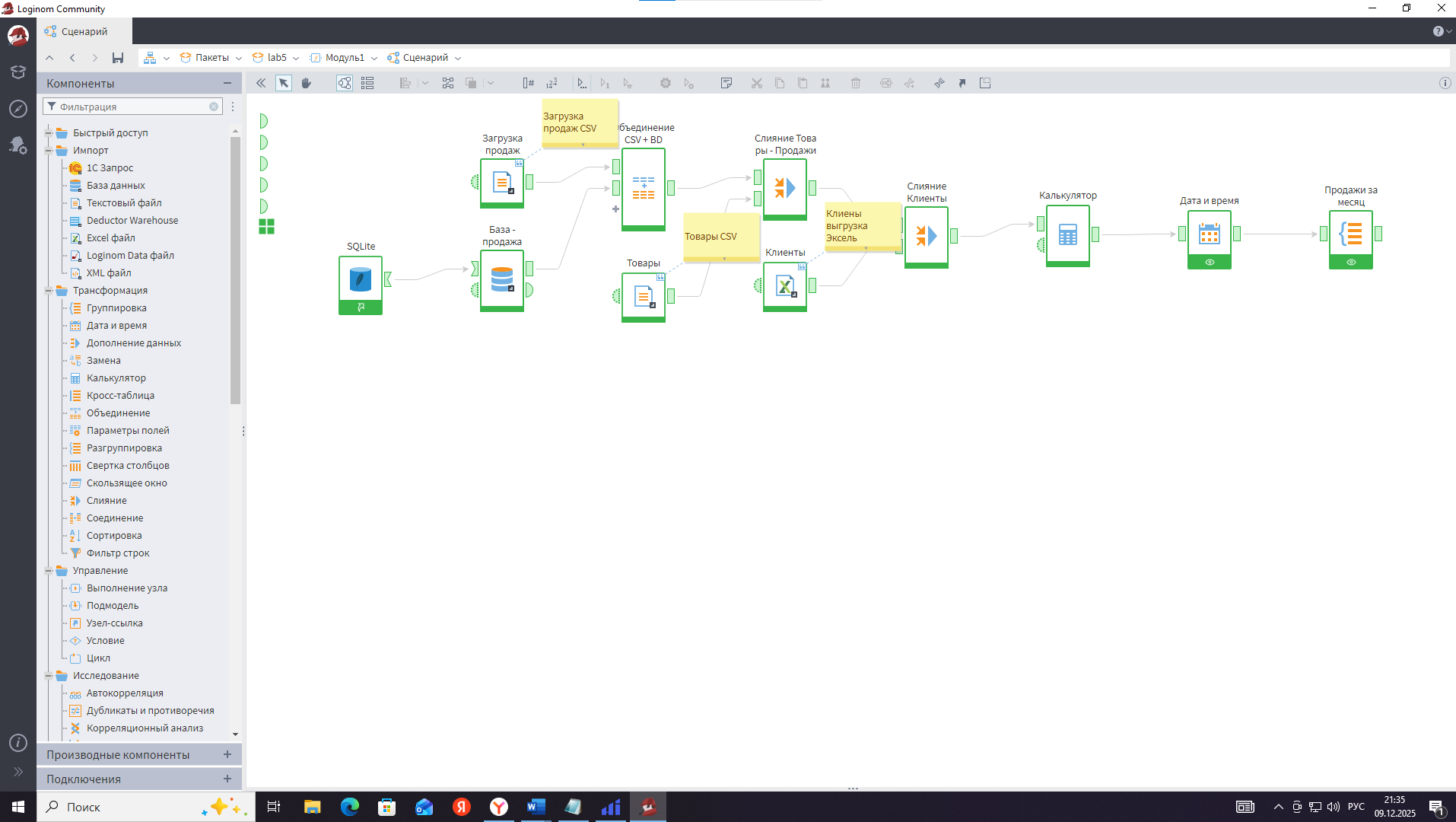
Лабораторная работа №5

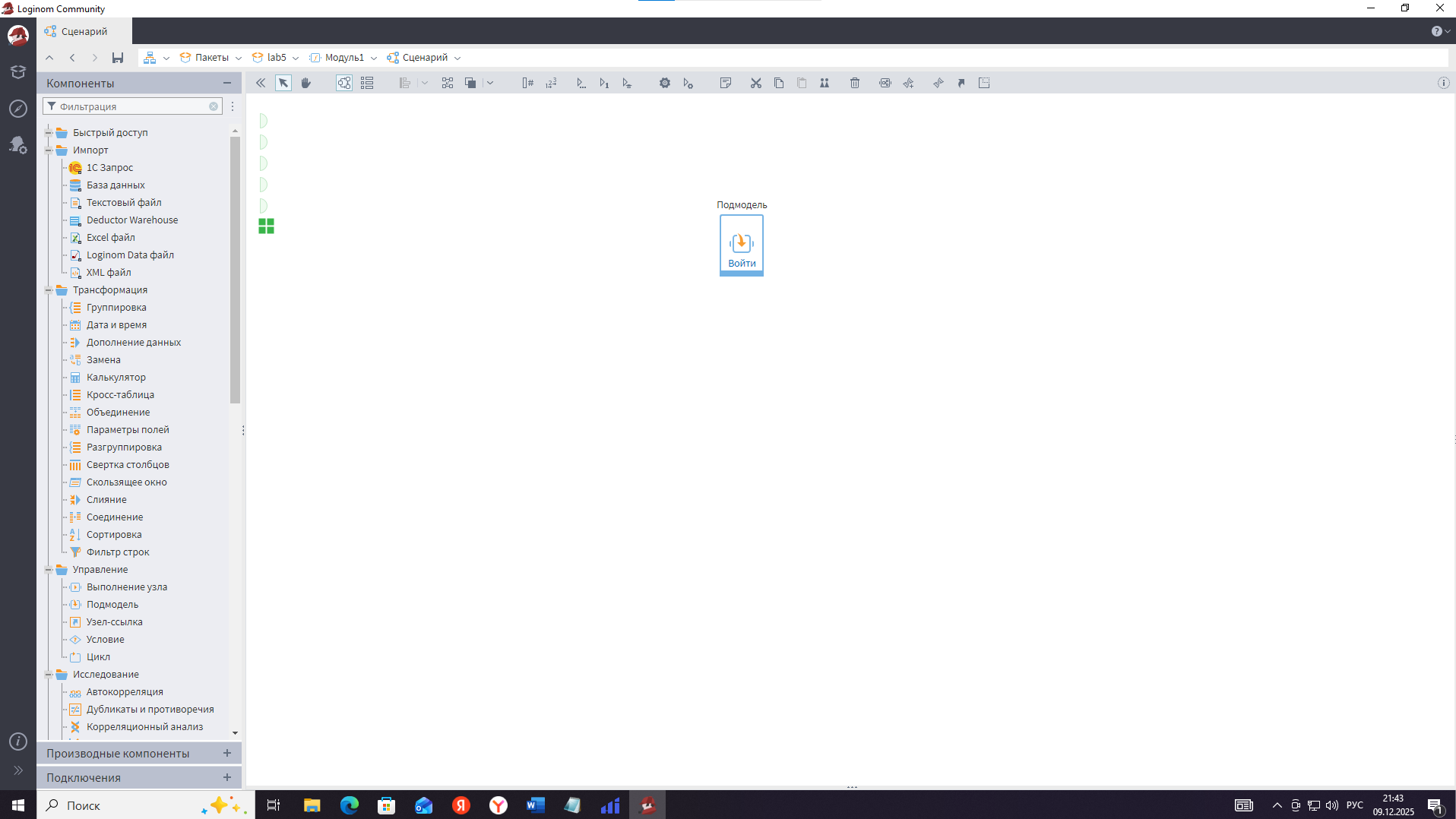
**Цель работы:** создать сценарий обработки данных о покупках клиентов для вычисления агрегированных показателей и формирования «портрета клиента» – то есть узнать о каждом клиенте больше, чем просто сумма его покупок.

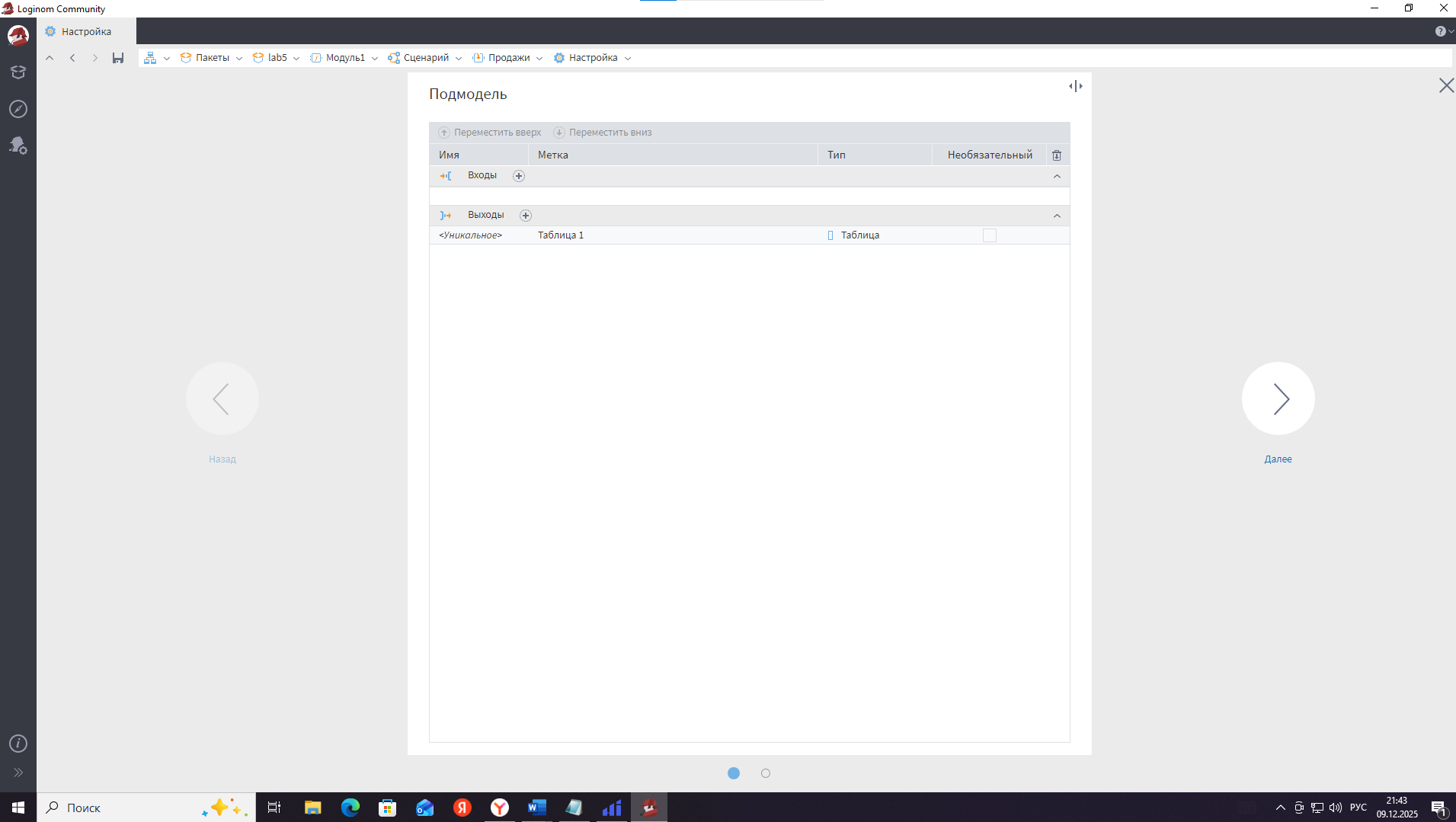
**Задачи:**

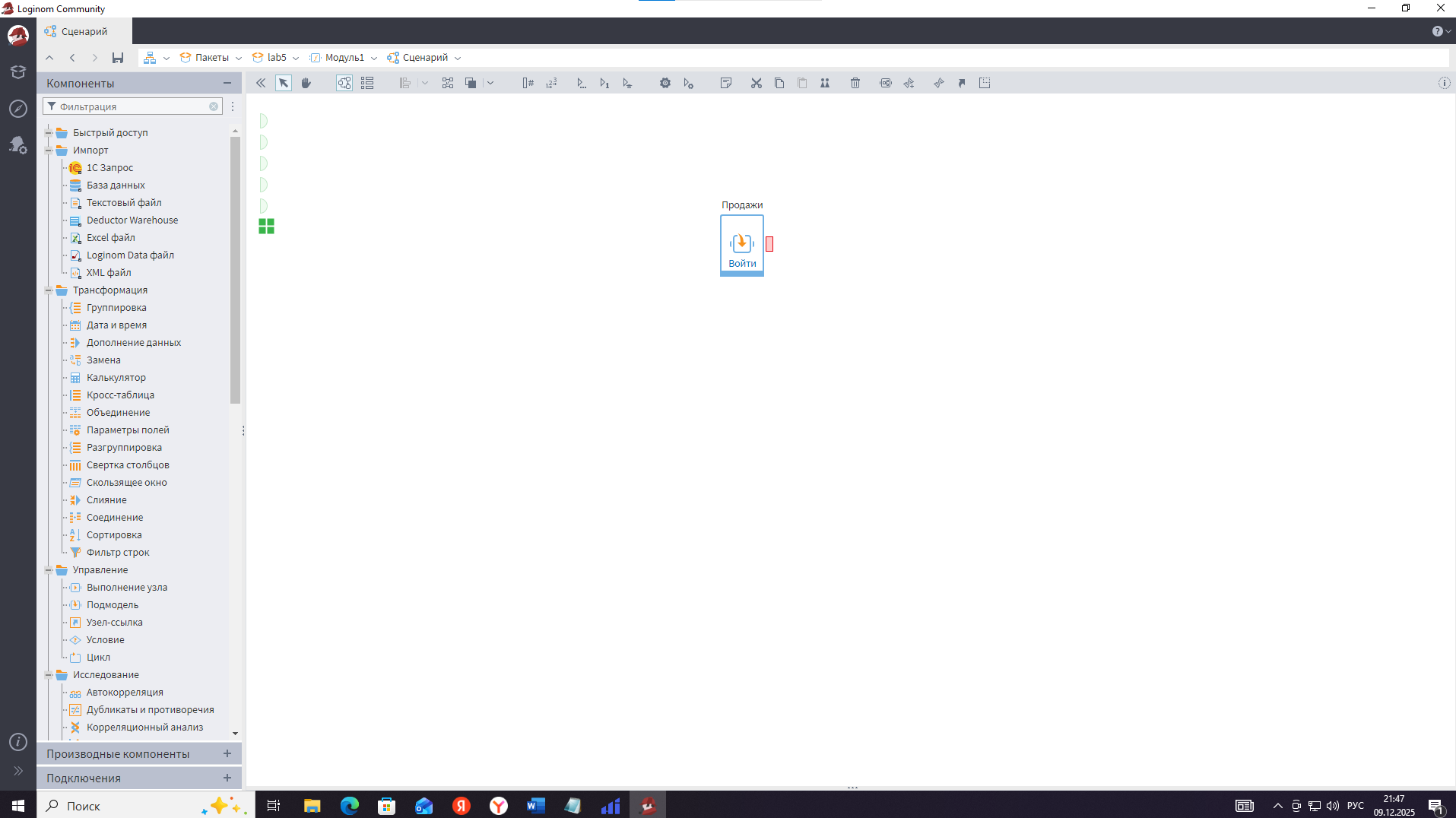
1. Подготовить копию пакета предыдущей лабораторной работы и привести сценарий к чистому состоянию (убрать лишние связи, обновить источники).
2. Подключить в сценарий таблицы продаж (из SQLite или файлов) и другие необходимые данные.
3. Свернуть всю ветку обработки «Продажи» в отдельную подмодель и настроить у неё входные/выходные порты.
4. С помощью узла «Группировка» построить агрегированные таблицы по клиентам (например, расчёт LTV – Lifetime Value).
5. Подключить готовые производные компоненты из библиотеки (например, «Профиль продаж» и «Любимые категории») и настроить их на имеющиеся данные.
6. Объединить различные результаты (профиль, категории, LTV) в единый справочник клиентов с помощью узла «Дополнение данных».
7. На базе полученного справочника создать дополнительные признаки через узел «Калькулятор»: например, Days\_from\_sale (сколько дней прошло с последней покупки), Client\_status (статус клиента), Client\_level (уровень – по объему покупок), Client\_activity (актуальность/активность клиента) и т.д.
8. Построить итоговые визуализаторы (кубы) «Сегментация клиентов» и «Портрет клиента» для аналитического представления результатов.
9. Протестировать сценарий на корректность, сохранить пакет и убедиться в целостности полученных данных.

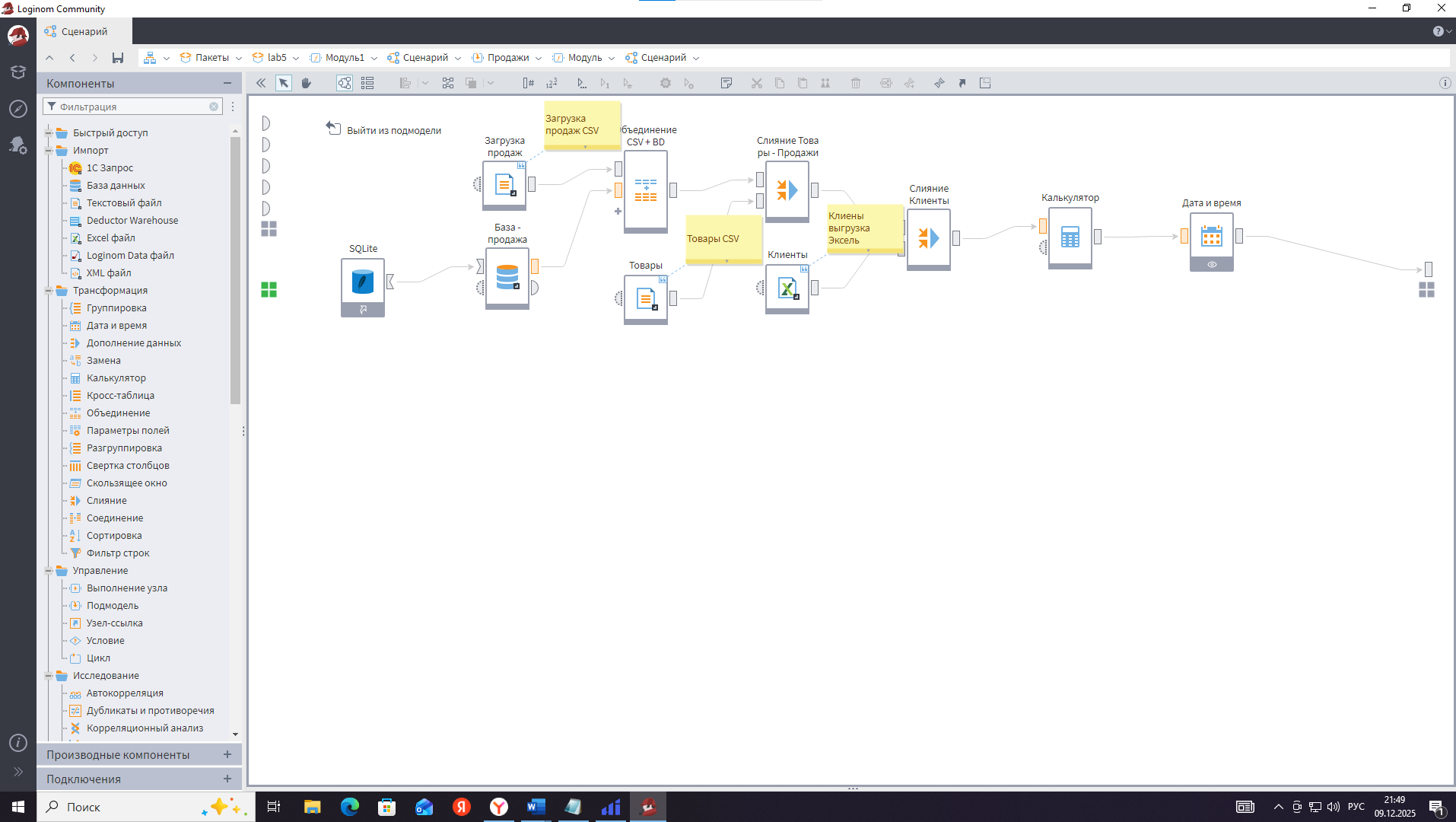
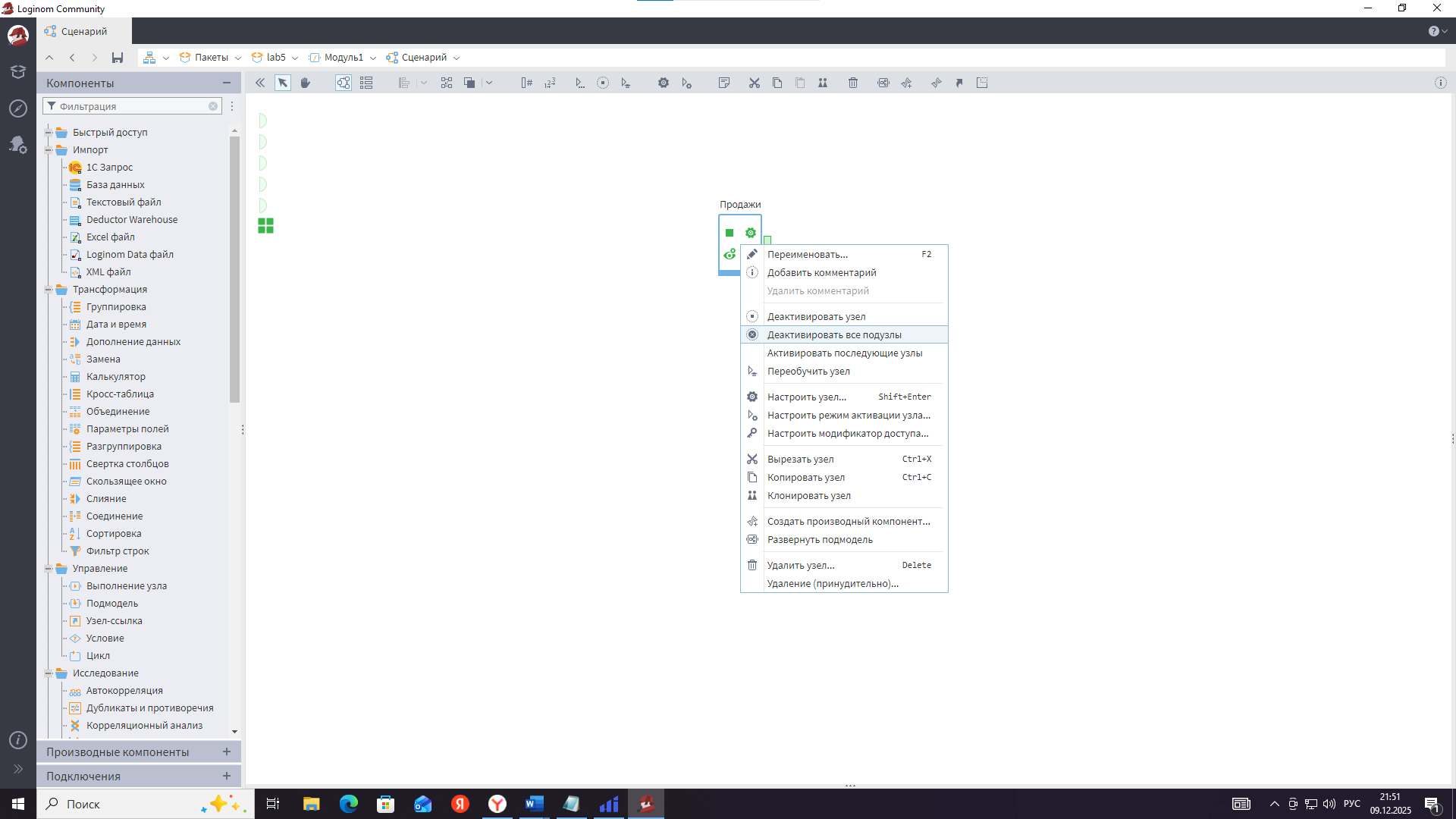
**Выполнение**

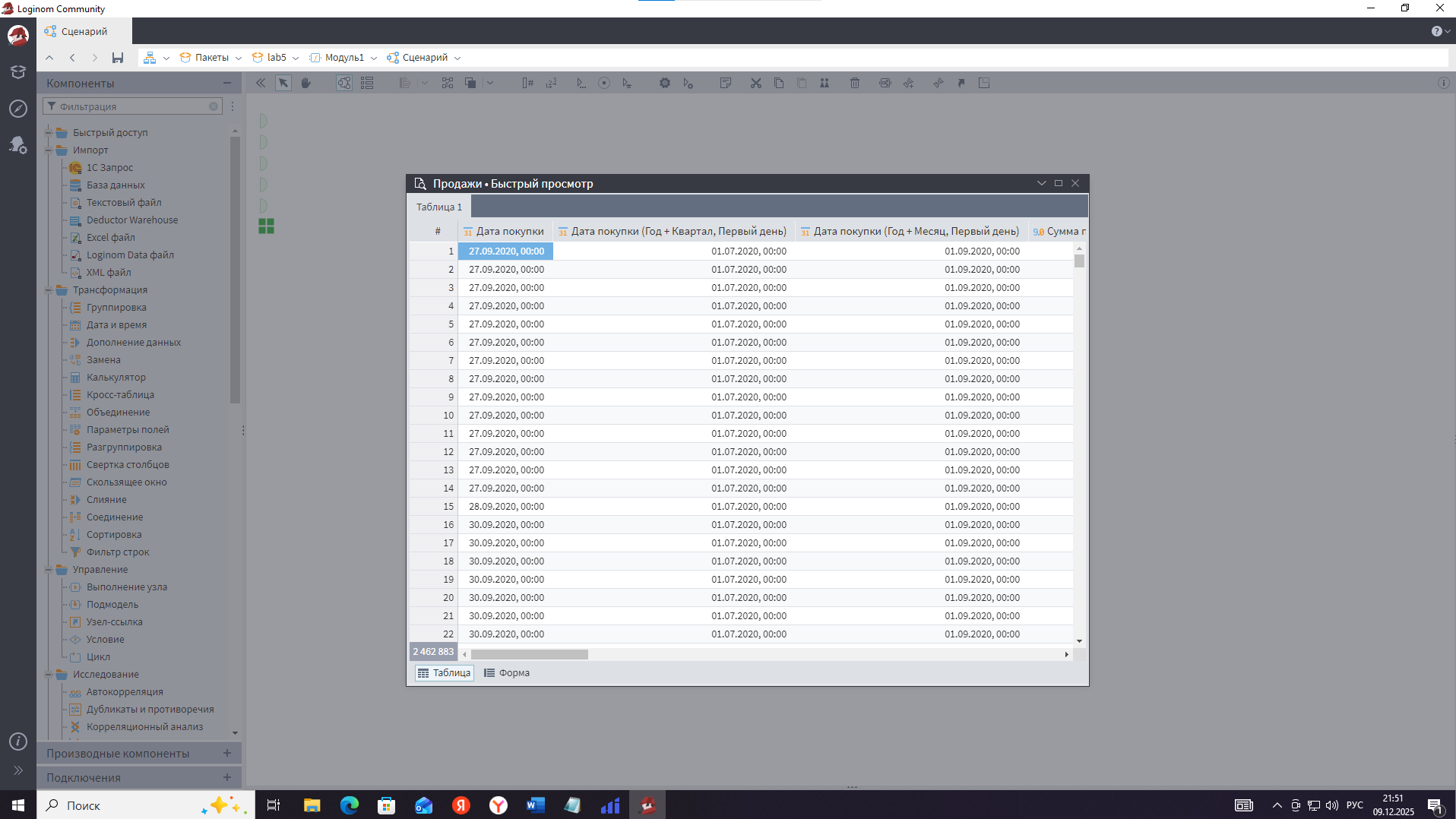
Подготовка рабочего пакета. Создаю отдельную папку «ЛР5\_Многомерный анализ» и помещаю в неё копию пакета, разработанного в ЛР4 (чтобы использовать уже интегрированные данные из БД и файлов) **Открываю этот пакет. Сценарий, оставшийся с прошлой работы, загружается без ошибок ** Прежде чем приступать, удаляю или исправляю все устаревшие связи: например, если пакет скопирован, надо убедиться, что пути к данным актуальны, или удалить лишние визуализаторы, которые могут мешать.

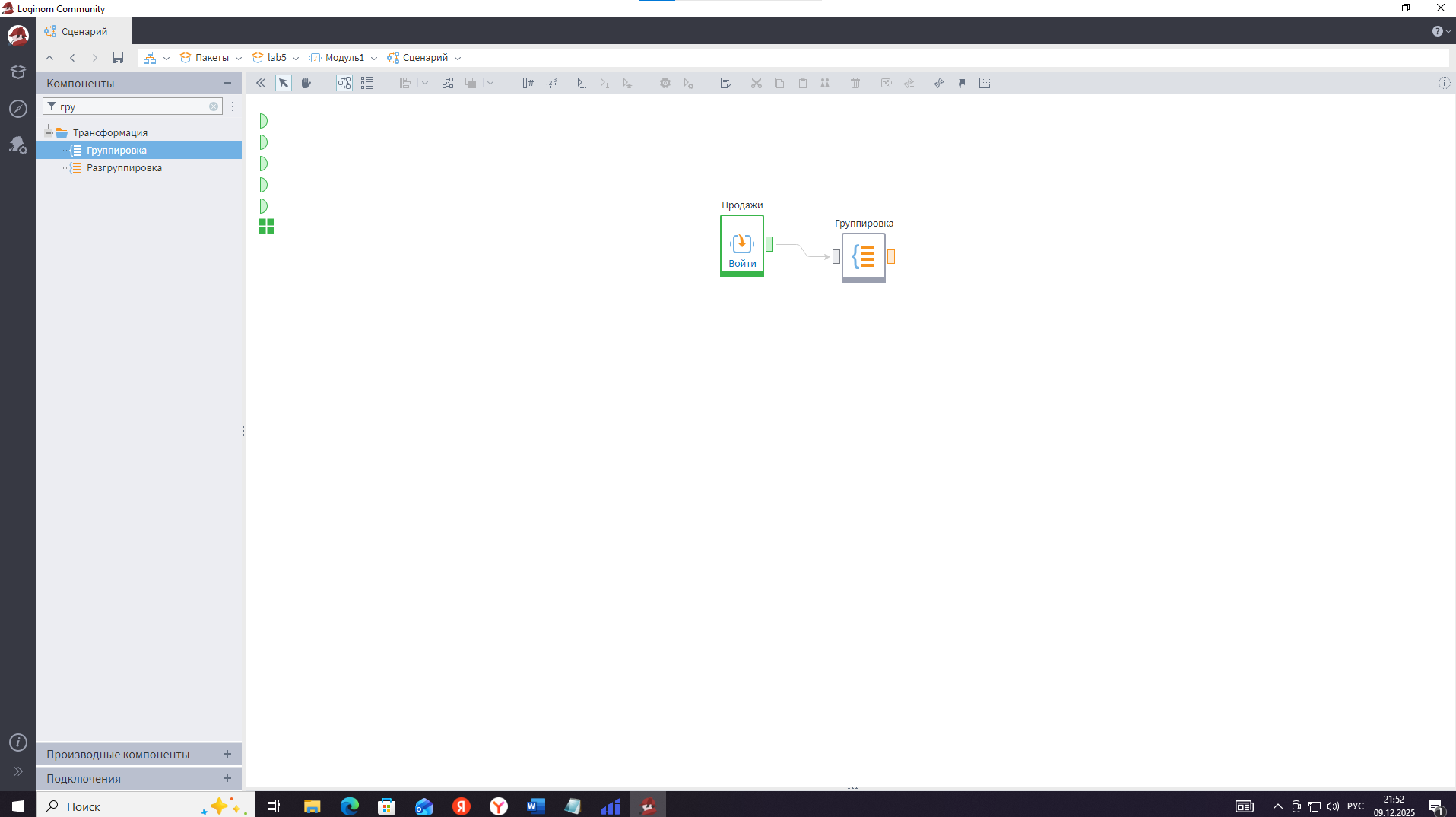
Организация ветки «Продажи» в подмодель. Основная часть сценария – обработка данных о продажах (загрузка, объединения, расчёты) – сейчас представлена множеством узлов. Для удобства их можно свернуть в один узел-подмодель. Выделяю на схеме все узлы, относящиеся к обработке транзакций продаж (от импорта данных до узла Дата и время, на котором происходят последние преобразования дат). Убедившись, что захвачены нужные элементы, на панели действий выбираю «Свернуть в подмодель». Loginom объединяет выделенные узлы в один контейнер **Называю эту подмодель «Продажи».

Теперь настраиваю подмодель. Кликаю по значку шестерёнки на узле Продажи. Появляются параметры подмодели, среди которых – настройка портов. Добавляю как минимум один табличный выходной порт, чтобы из подмодели можно было передать наружу получившиеся данные. Назначение этого порта – выдавать список всех транзакций продаж (возможно, с рассчитанными доп. полями). Присваиваю порту понятное имя, например «Транзакции». Сохраняю настройки **Теперь у узла Продажи появился выход (сначала он красный, так как внутри ничего к нему не подключено).

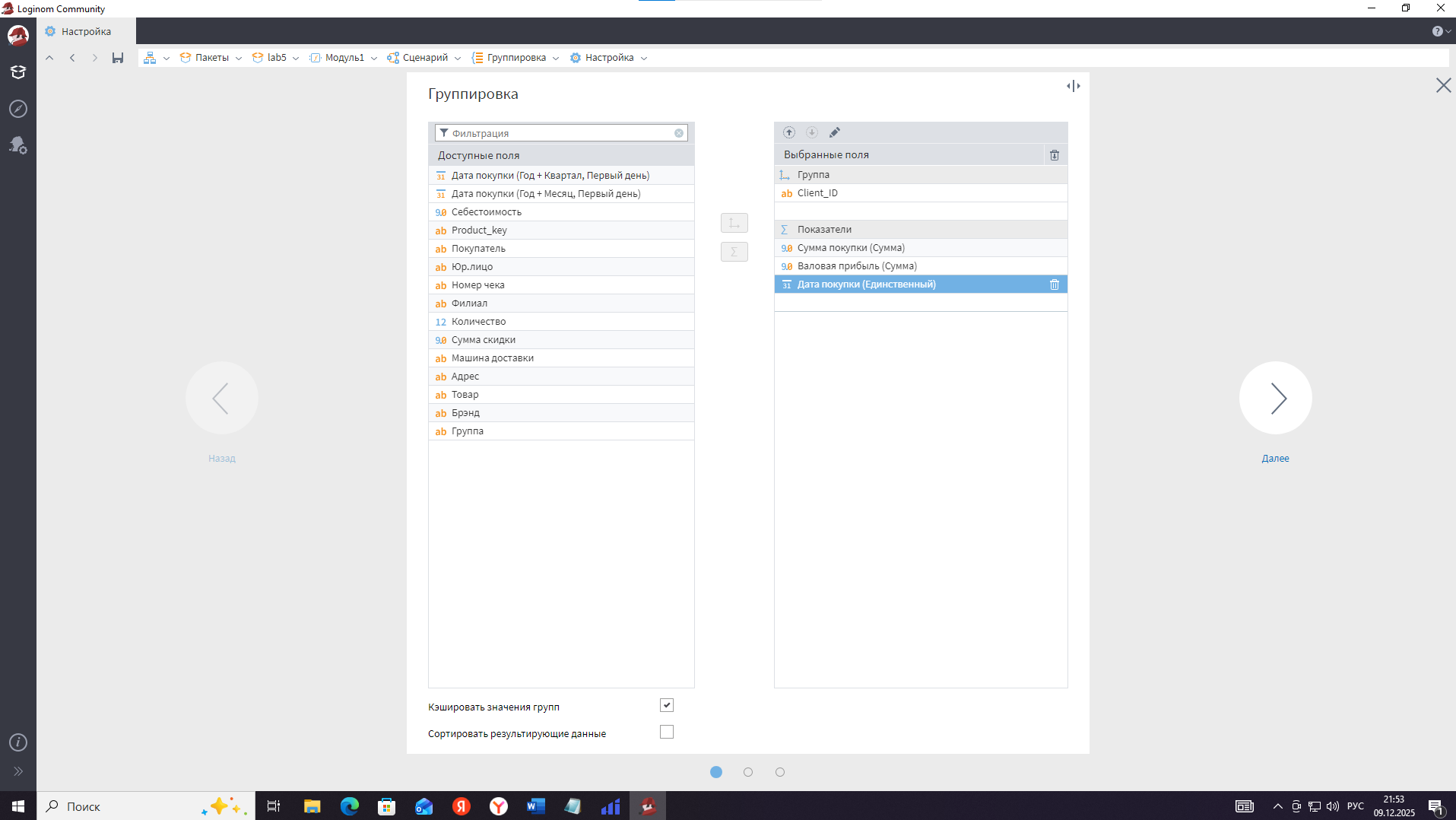
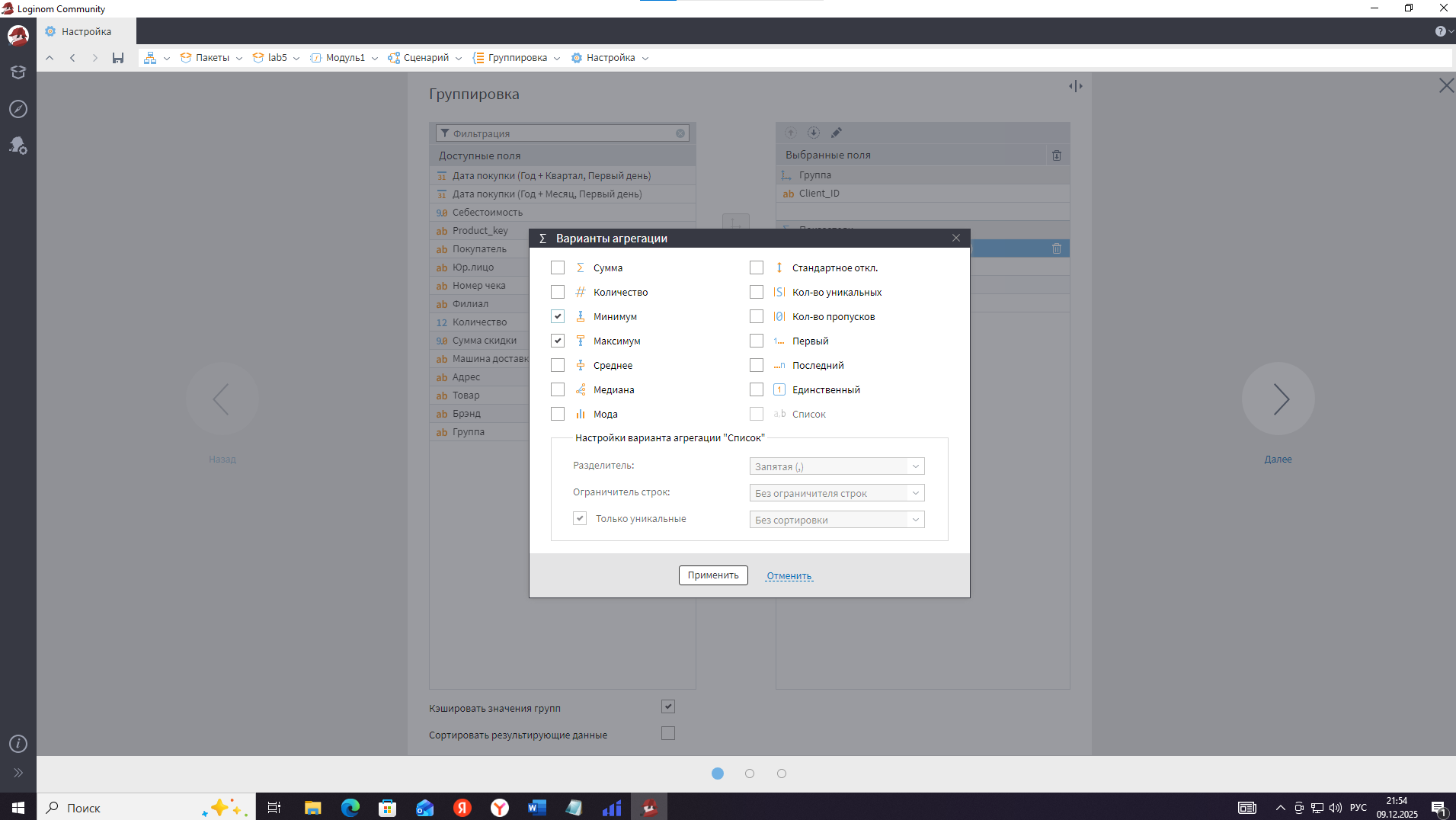


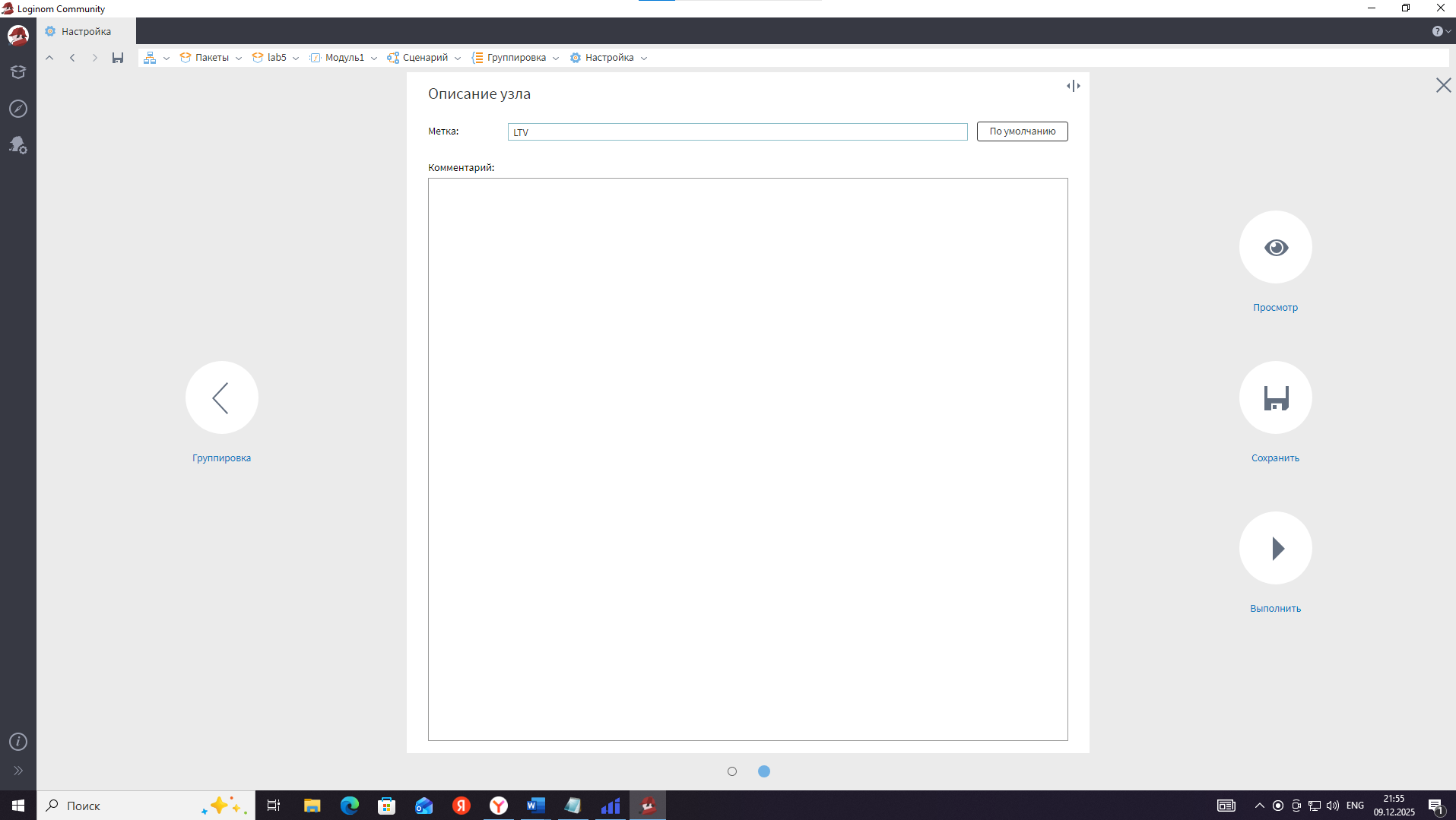
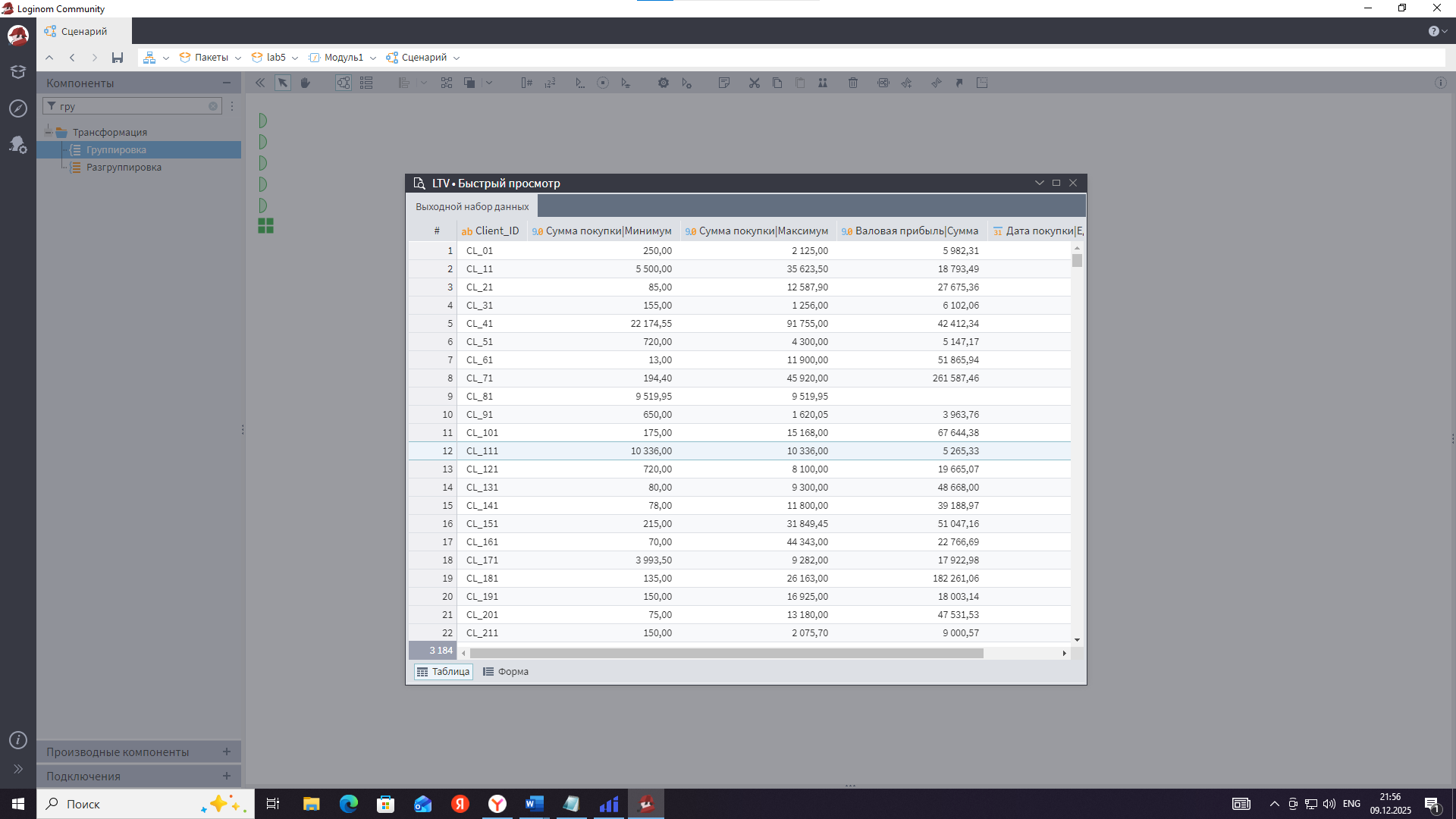
Двойным кликом вхожу внутрь подмодели «Продажи». Вижу там все свернутые узлы. Нужно связать внутренний поток данных с выходным портом. Решаю, что отдавать наружу буду данные после узла Дата и время (это уже обогащенные транзакции с валовой прибылью и периодами). Поэтому внутри подмодели соединяю выход узла *Дата и время* с портом Транзакции **Выхожу обратно на верхний уровень. Теперь внешний узел Продажи имеет активный выход (не красный). Чтобы убедиться, что всё работает, запускаю заново выполнение подмодели: либо изнутри, либо на внешнем уровне нажимаю «Выполнить» на узле Продажи, а также активирую все узлы, которые идут после него **

Проверяю содержимое: открываю предпросмотр на выходе подмодели Продажи – отображается таблица всех продаж с теми же колонками, что были внутри (дата, филиал, товар, сумма, себестоимость, валовая прибыль, производные периоды и т.д.) **Количество записей должно соответствовать общему числу транзакций, что и было ранее.

Расчет LTV по клиентам (агрегация по клиенту). Теперь на основе транзакционных данных посчитаем совокупные показатели для каждого клиента (Lifetime Value – суммарная выручка и прибыль, число покупок, даты первой и последней покупки). Сразу после узла Продажи (подмодели) добавляю узел «Группировка» и соединяю его вход с выходом Продажи **Открываю настройки группировки. В разделе «Группировать по» указываю поле Client\_ID – чтобы получить одну строку на каждого клиента. В разделе «Показатели» выбираю следующие поля:

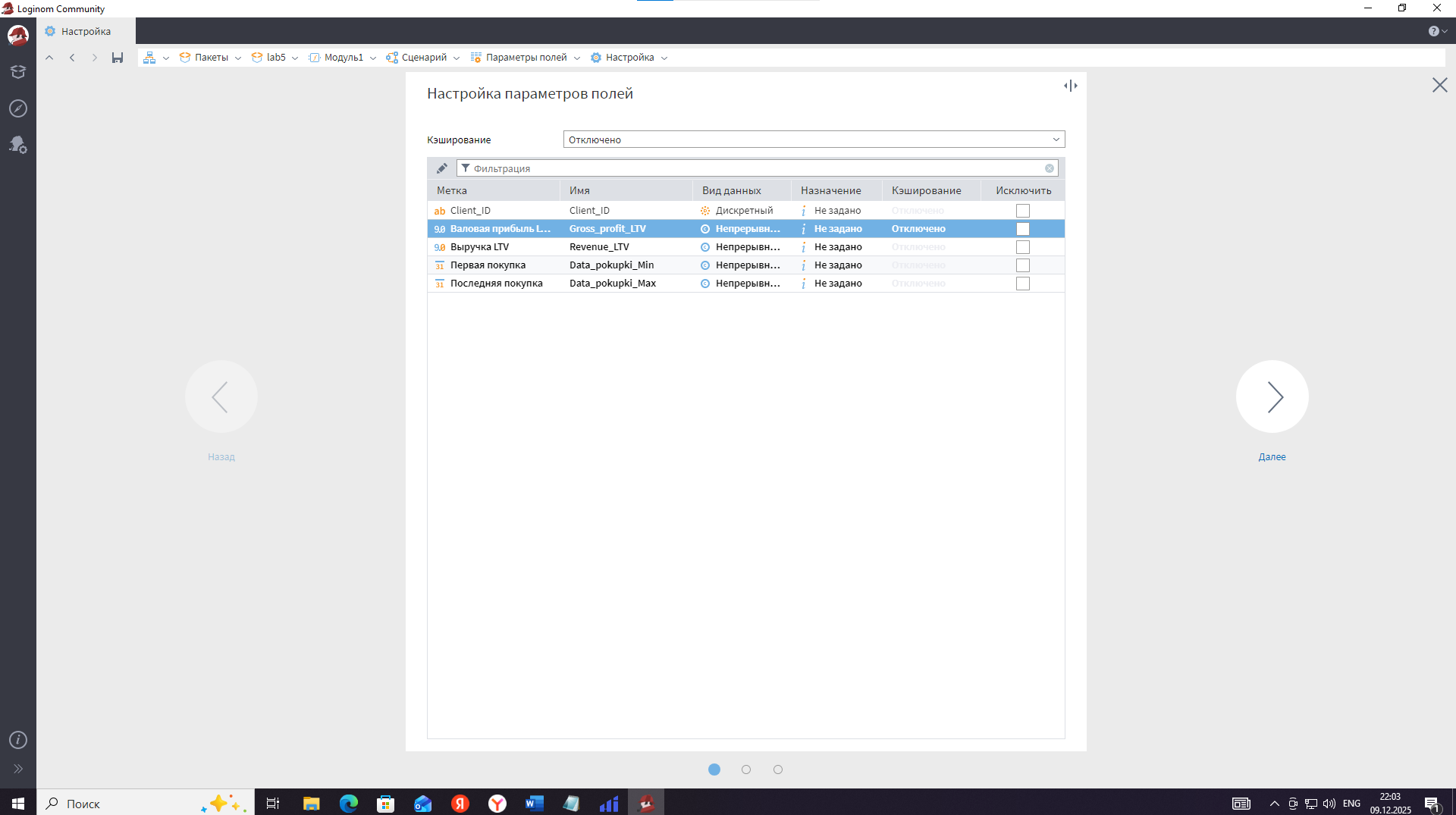
1. *Сумма покупки* – агрегирование Сумма (итоговая выручка от клиента).
2. *Валовая прибыль* – агрегирование Сумма (итоговая прибыль от клиента).
3. *Дата покупки* – здесь нужно получить мин. и макс. даты.

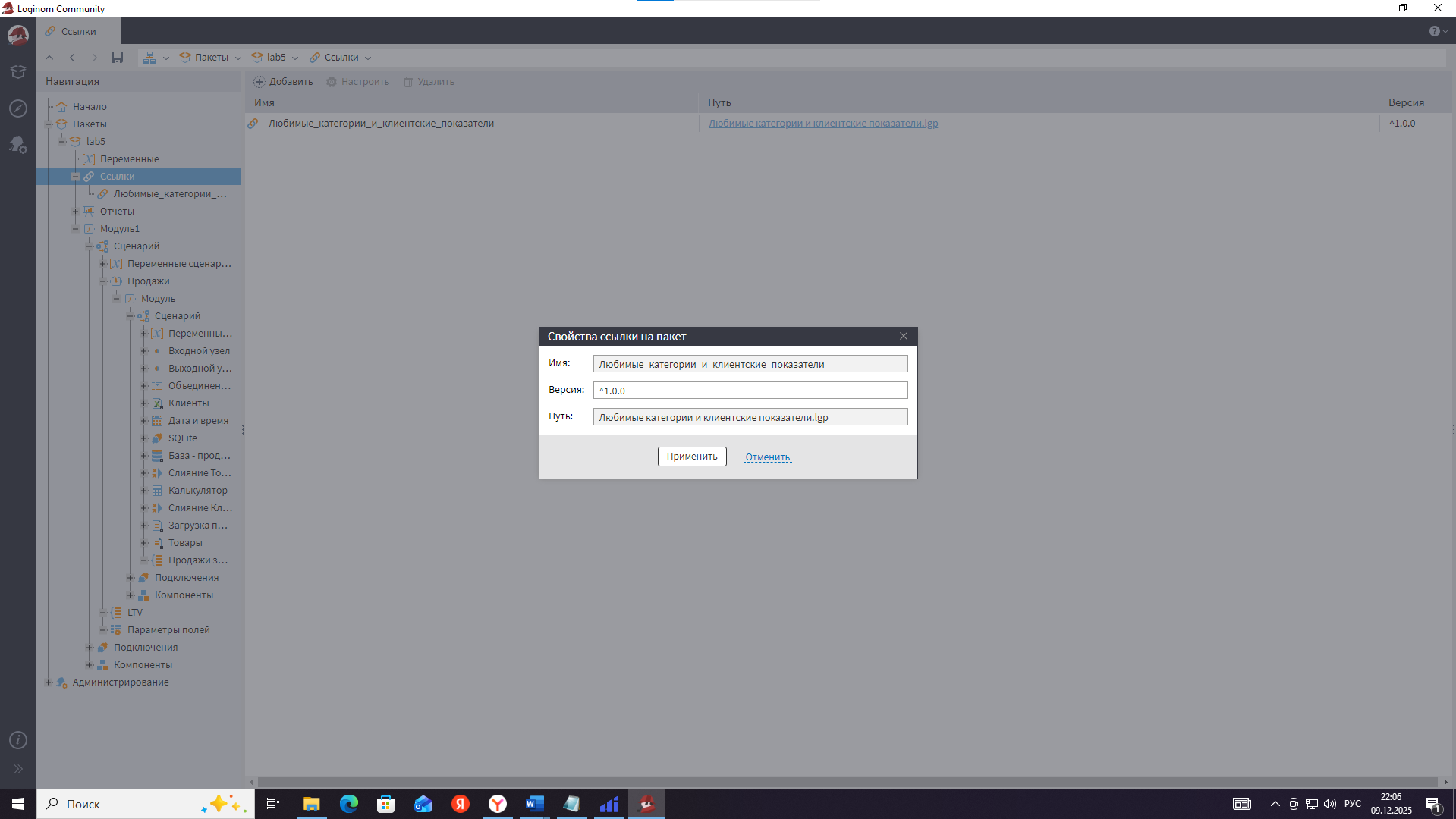
**В мастере сначала добавляю это поле один раз (по умолчанию агрегирование Сумма смысла не имеет для даты), поэтому дважды кликаю по добавленному показателю и изменяю метод агрегации: выбираю Минимум (получим дату первой покупки). Затем добавляю *ещё раз* поле Дата покупки и для второго экземпляра выбираю Максимум (дата последней покупки) 

Переименовываю узел группировки в LTV, чтобы было ясно, что он формирует показатели “всей жизни” клиента ** Активирую (выполняю) его. В результате появляется таблица, в которой каждая строка – уникальный клиент, а колонки содержат: сумму всех его покупок, суммарную валовую прибыль, дату первой и последней сделки. Для контроля смотрю на количество строк – их 3184, что соответствует числу клиентов в исходных данных 

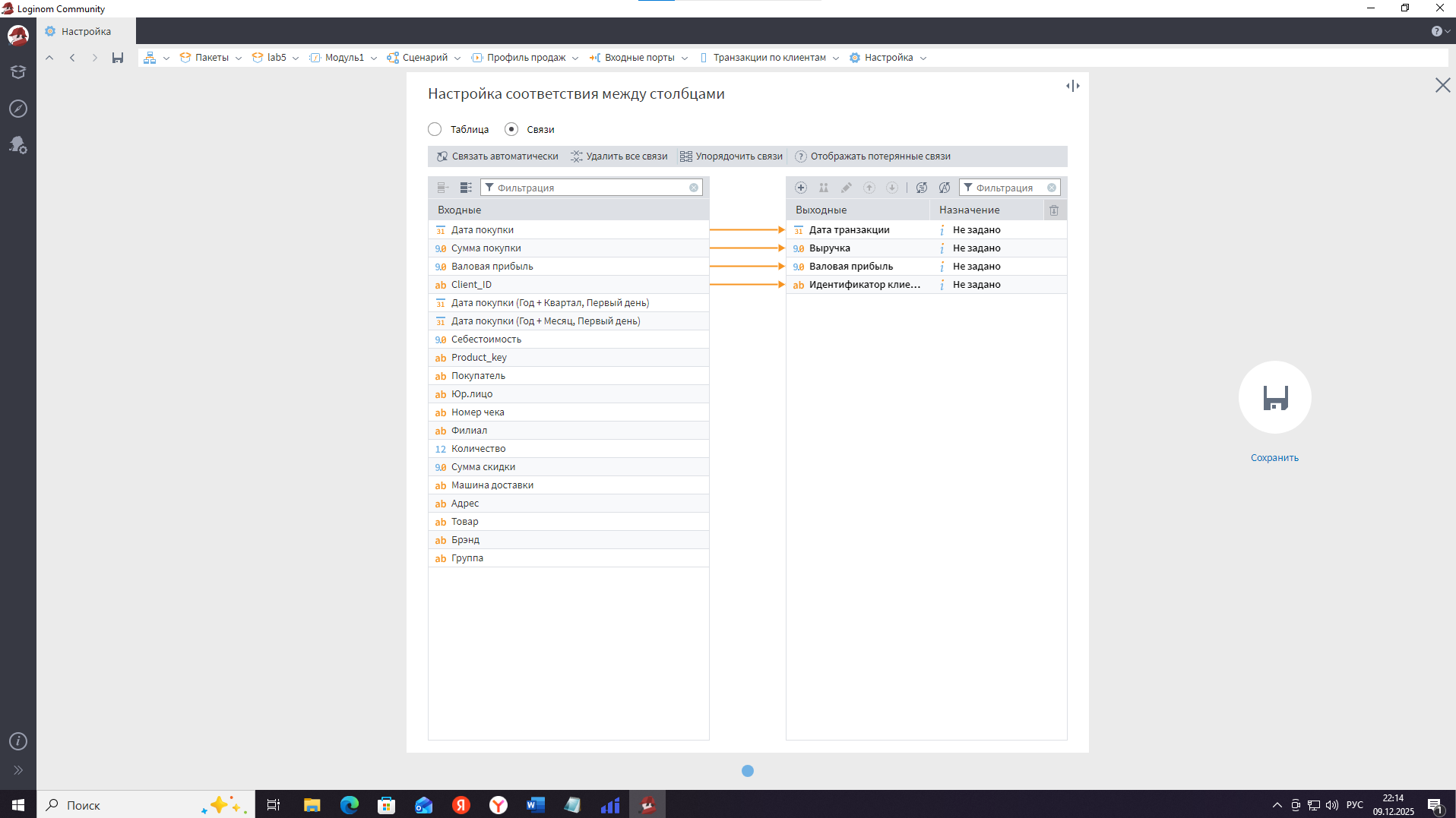
Приведение имен полей в удобный вид. В таблице LTV, полученной после группировки, названия полей сформированы автоматически (например, SUM(Sum\_purchase), MIN(Date\_purchase) и т.д.). Для удобства переименую их. Добавляю узел «Параметры полей» и соединяю: выход LTV -> вход Параметры полей. В настройках узла вижу список полей и их текущие имена/метки. Делаю так:

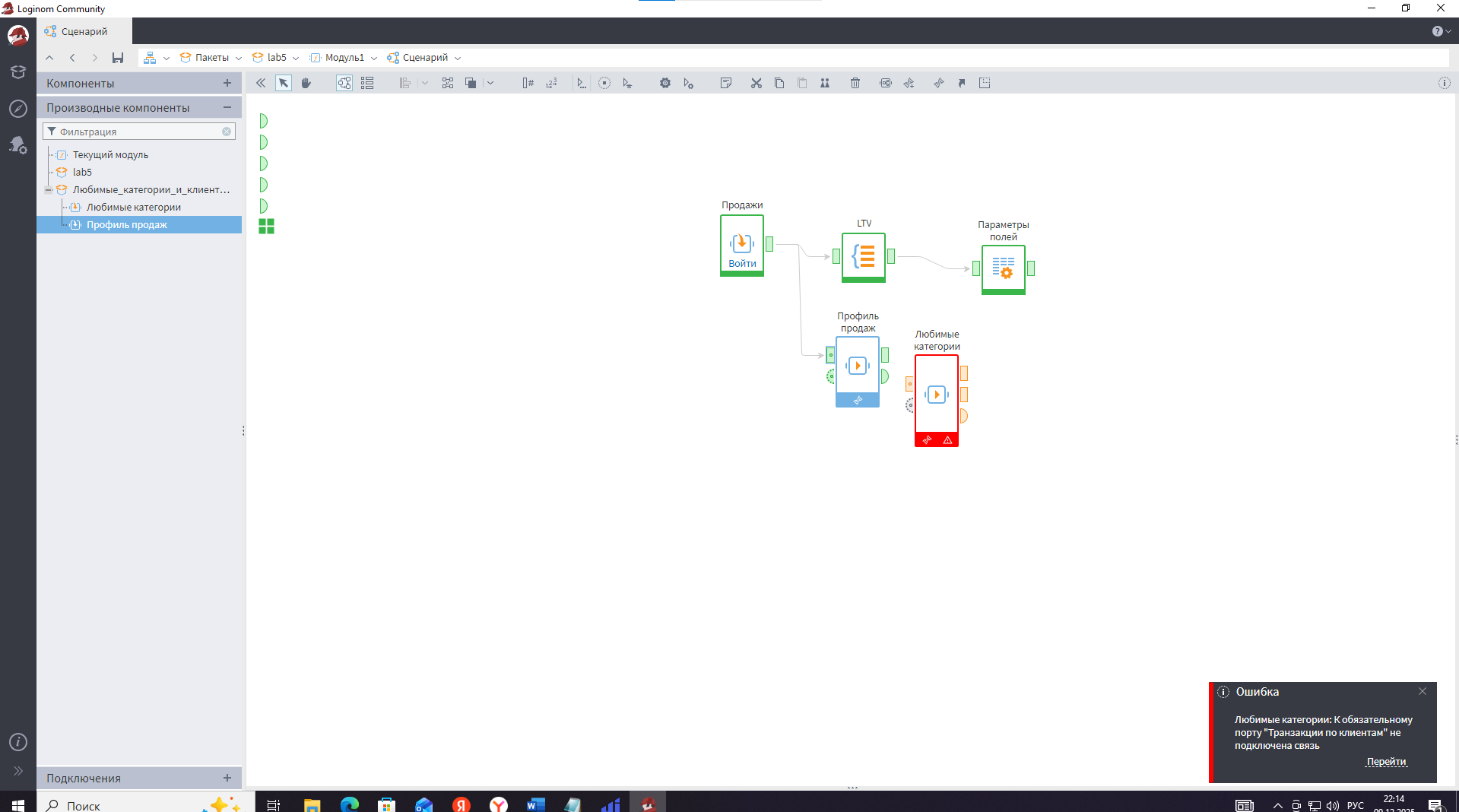
1. Поле SUM(Sum\_purchase) переименовываю: Имя = Revenue\_LTV, Метка = «LTV выручка».
2. Поле SUM(Gross\_profit) -> Имя = Gross\_profit\_LTV, Метка = «LTV валовая прибыль».
3. Поле MIN(Date\_purchase) -> Имя = First\_purchase\_date, метка = «Первая покупка».
4. Поле MAX(Date\_purchase) -> Имя = Last\_purchase\_date, метка = «Последняя покупка».

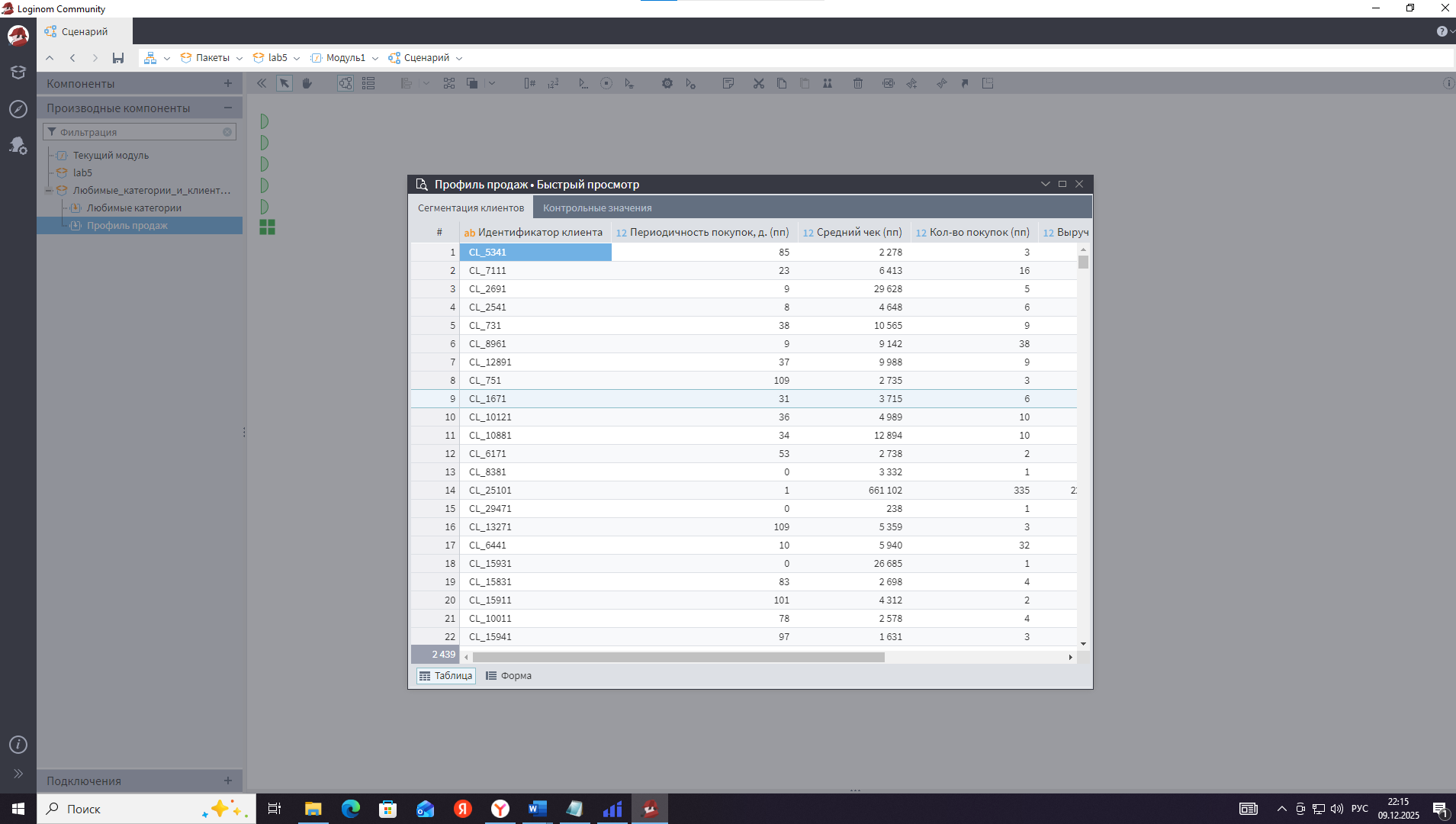
** Сохраняю изменения. Теперь таблица LTV имеет удобные заголовки колонок, и в дальнейшем мне будет проще ссылаться на них в формулах.

Подключение производных компонентов: «Профиль продаж» и «Любимые категории». В Loginom существует библиотека готовых решений (компонентов), которую можно подключить к проекту. В данном случае полезными будут компоненты для анализа клиентской активности и их предпочтений. Открываю раздел «Ссылки» пакета, нажимаю «Добавить» и выбираю пакет библиотеки, содержащий нужные компоненты (например, стандартную библиотеку от Loginom, где есть «Профиль продаж клиента» и «Любимые категории»). После добавления ссылки в панели компонентов появляется раздел «Производные компоненты» с соответствующими узлами **

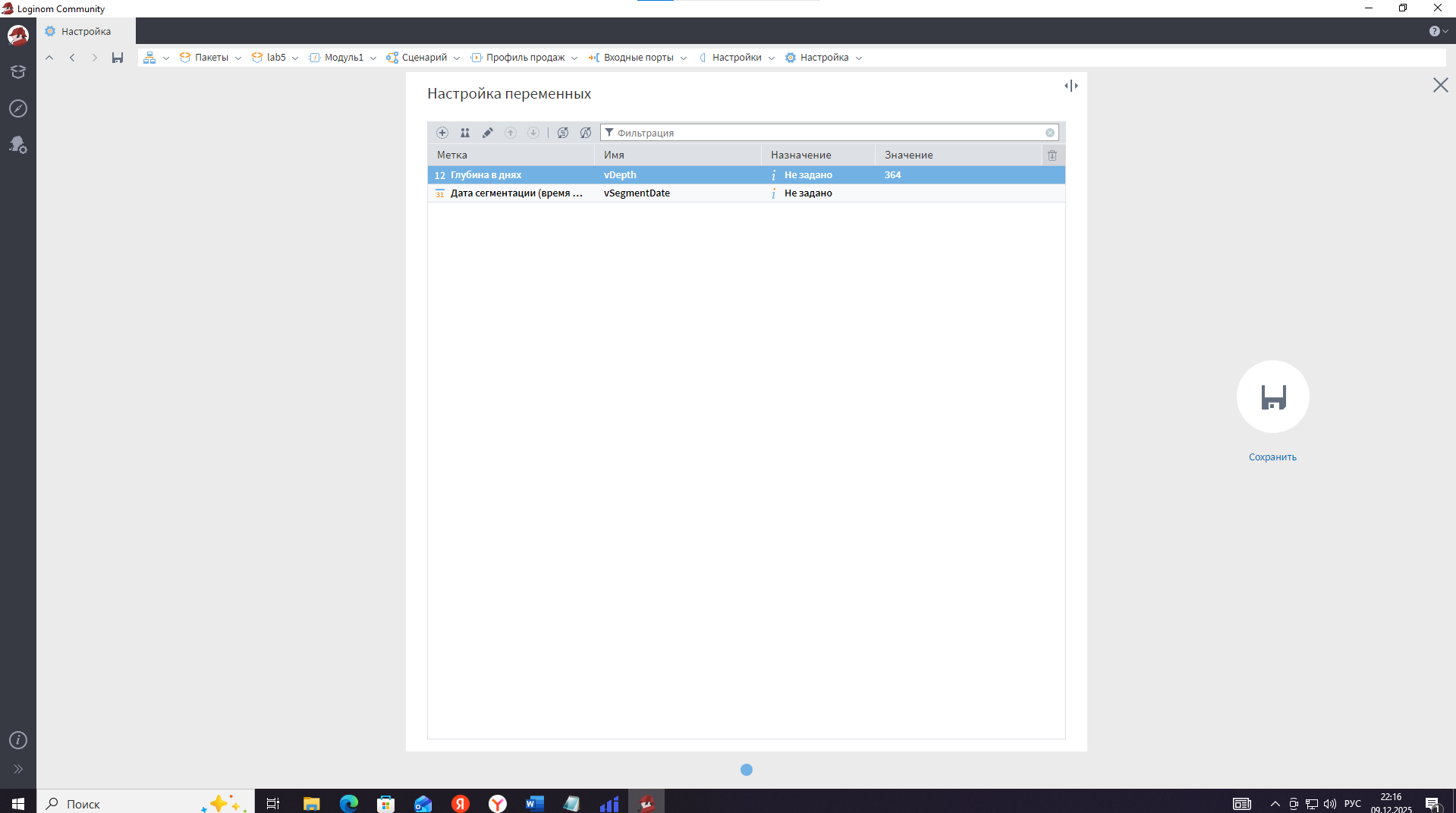
Перетаскиваю на схему узел «Профиль продаж» и узел «Любимые категории» **Эти компоненты представляют собой подмодели, требующие настройки входов.

Сначала настрою «Профиль продаж». Соединяю его вход Табличные данные с выходом подмодели Продажи (чтобы компонент получил все транзакции клиентов). При соединении автоматически открывается окно сопоставления полей: необходимо связать поля из нашей таблицы транзакций с теми, которые ожидает компонент. Обычно по именам система сама многое сопоставляет, но я проверяю: Client\_ID->Client\_ID, Дата покупки->Date, Сумма покупки->Amount, Валовая прибыль->Profit, ну и так далее для всех ключевых полей. Нажимаю кнопку «Упорядочить связи» для автоматического сопоставления и вношу правки, если что-то не совпало **Подтверждаю и закрываю это окно. Затем дважды кликаю по входу Переменные узла «Профиль продаж». Тут можно установить параметры расчета профиля: Дата сегментации (на какую дату считать показатели клиента) и Глубина в днях (за сколько дней истории учитывать, по умолчанию 365). Оставляю дату пустой (значит, возьмётся самая последняя дата из данных как точка отсчёта) и глубину ставлю 364 (расчет за год) Закрываю настройки переменных.

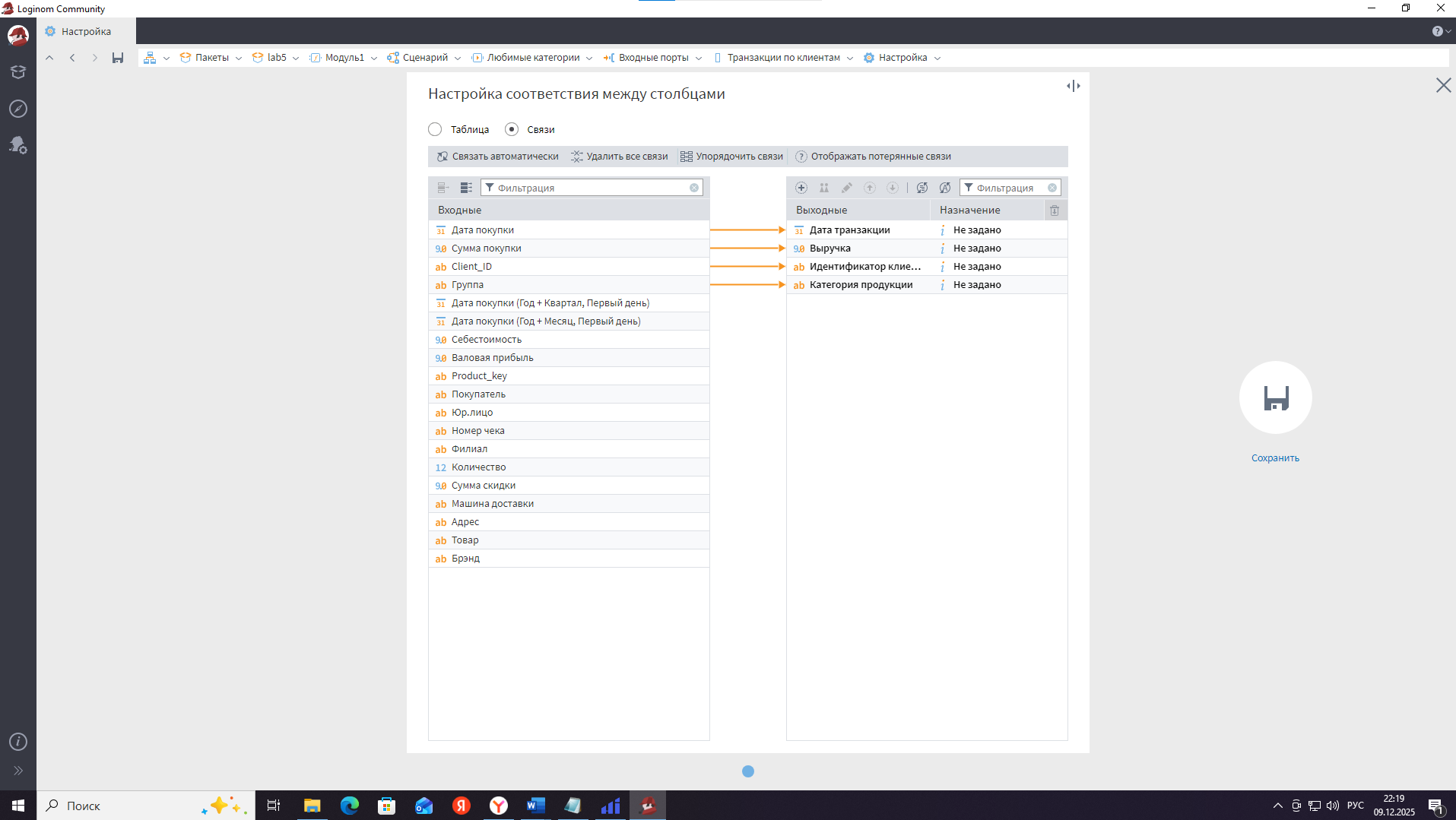
Активирую узел Профиль продаж. После выполнения он становится зелёным **На выходном порту Табличные данные этого узла должна появиться таблица, где на каждого клиента рассчитаны детальные показатели: число покупок, средний интервал между покупками, средний чек, общая выручка, общая валовая прибыль, доля валовой прибыли в обороте и т.д. Открываю предпросмотр – действительно вижу строки с Client\_ID и набором метрик

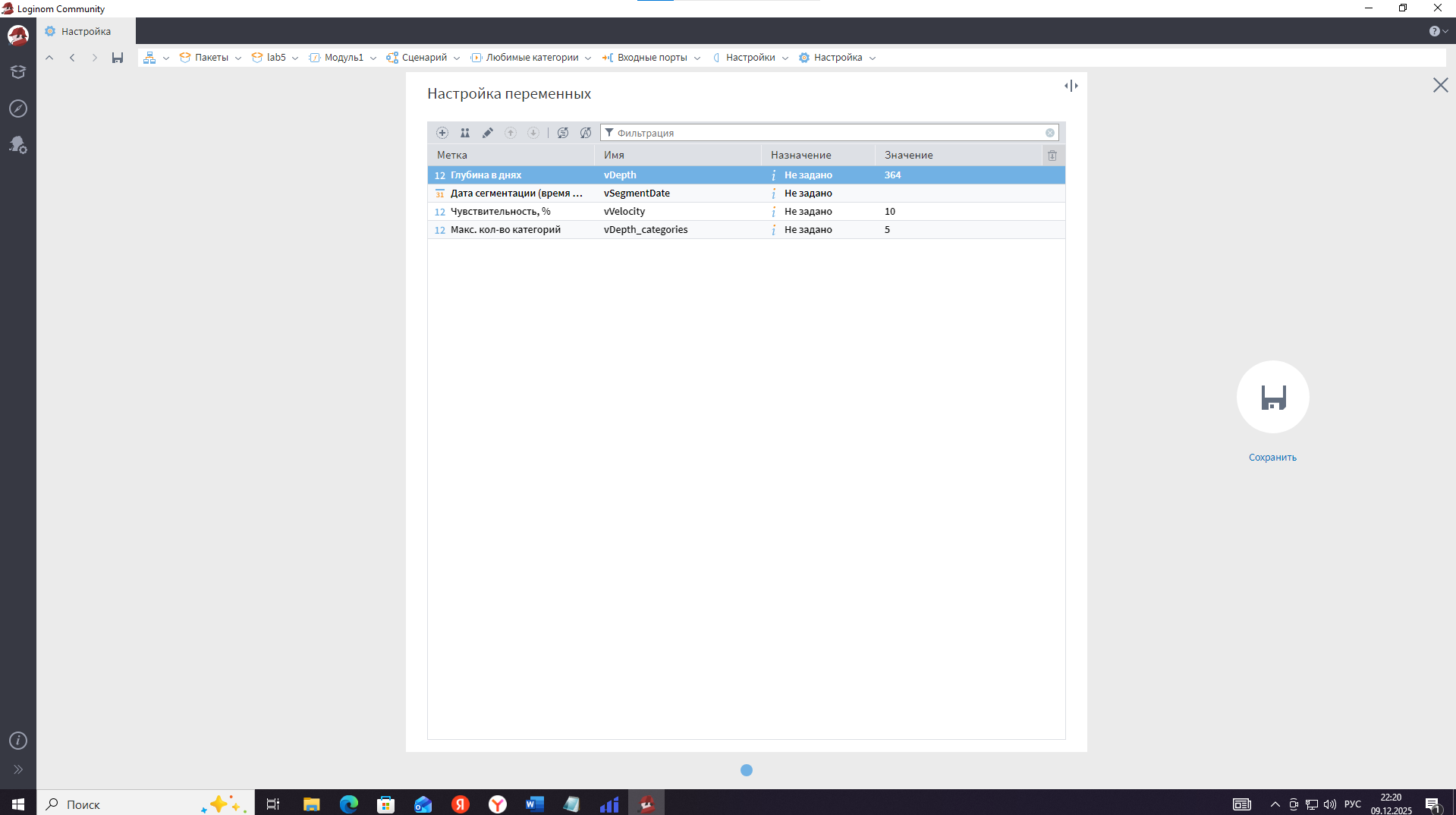
**

Эти данные отражают поведение клиентов за последний год (364 дня) на текущую дату.



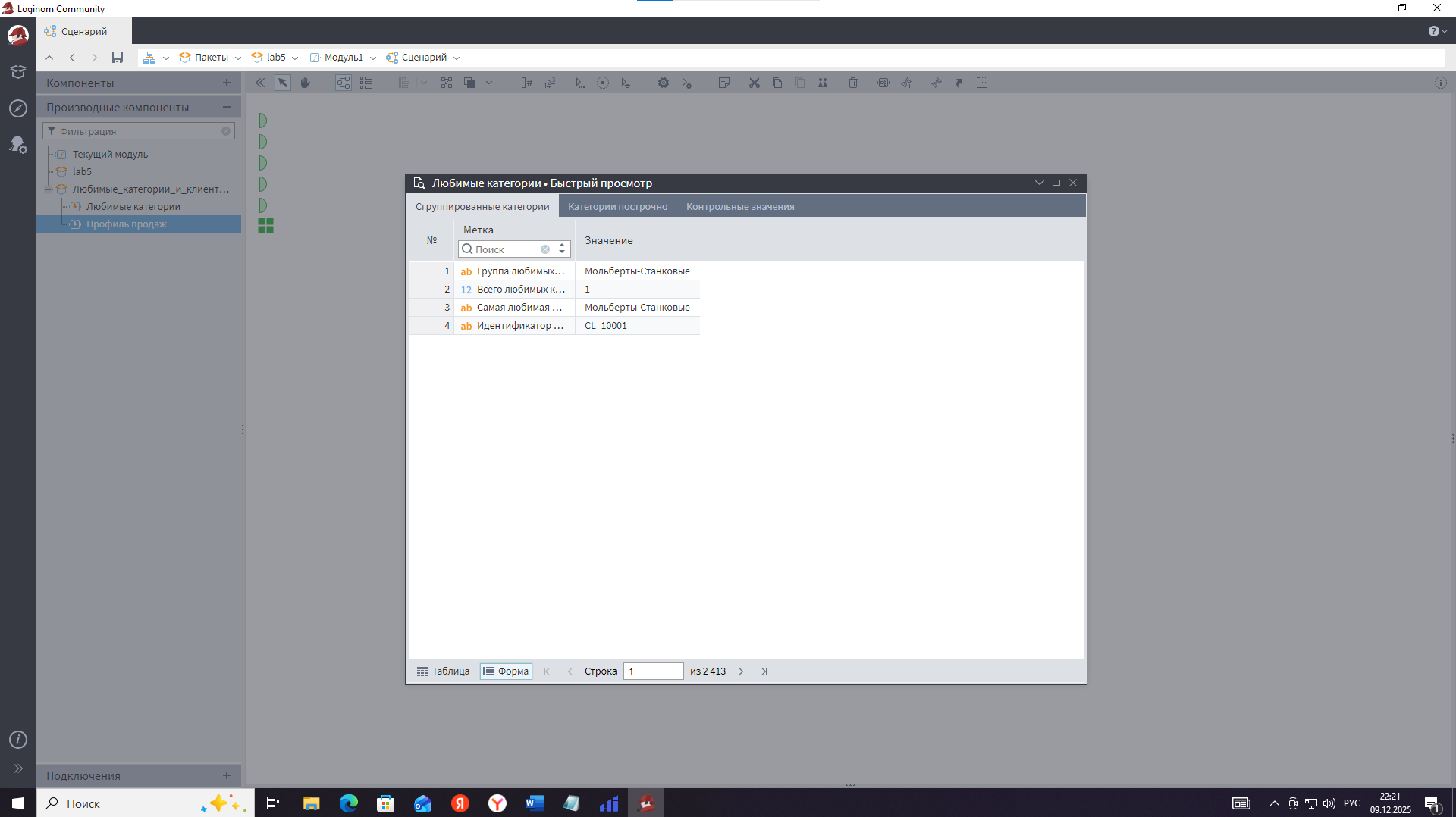


Теперь настраиваю «Любимые категории». Соединяю его вход Табличные данные также с выходом подмодели Продажи (ему тоже нужны все транзакции по клиентам). Открывается аналогичное окно сопоставления: связываю идентификатор клиента, поле категории товара, сумму покупки – то есть поля, необходимые для анализа любимых категорий. В моем случае поле *SKU\_group\_name* (или другое, обозначающее категорию товара) связываю с ожидаемым полем Category, Client\_ID -> Client\_ID, Сумма -> Amount и т.п. **Затем двойной щелчок по входу Переменные узла «Любимые категории» – ввожу параметры: Дата сегментации (аналогично, оставляю пустой для максимальной даты), Глубина – 364 дня, Чувствительность, % – оставляю значение по умолчанию (это порог доли от покупок, выше которого категория считается «любимой»), Макс. кол-во категорий – можно указать, например, 3 (будем выделять до трех любимых категорий на клиента)

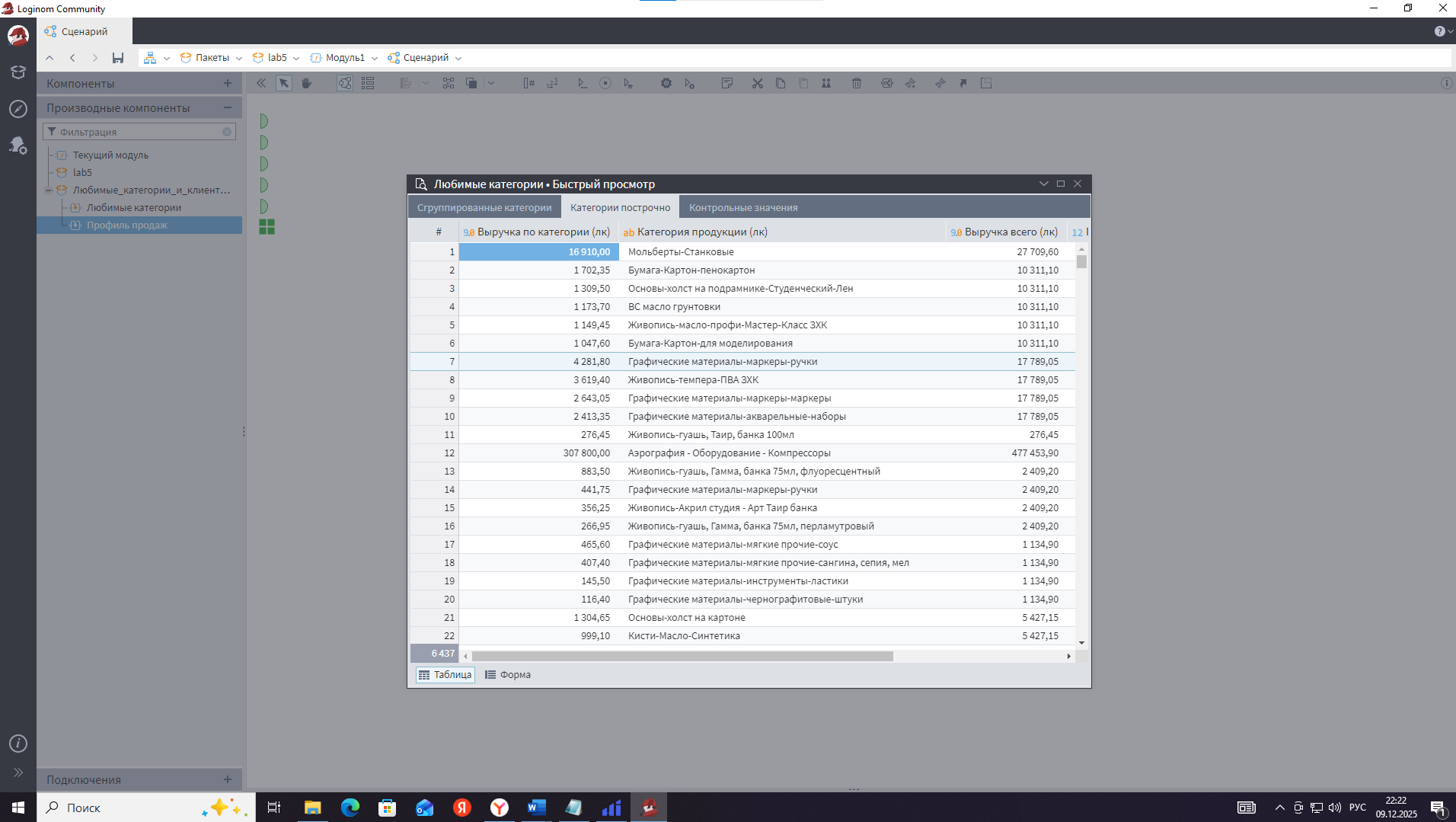
Выполняю узел Любимые категории. Этот компонент имеет два выхода: - «Таблица»: в свернутом виде для каждого клиента перечислены его любимые категории (например, через запятую или множеством столбцов). - «Форма»: развернутые данные – каждая запись это пара клиент+его любимая категория (в случае если у клиента несколько фаворитов).

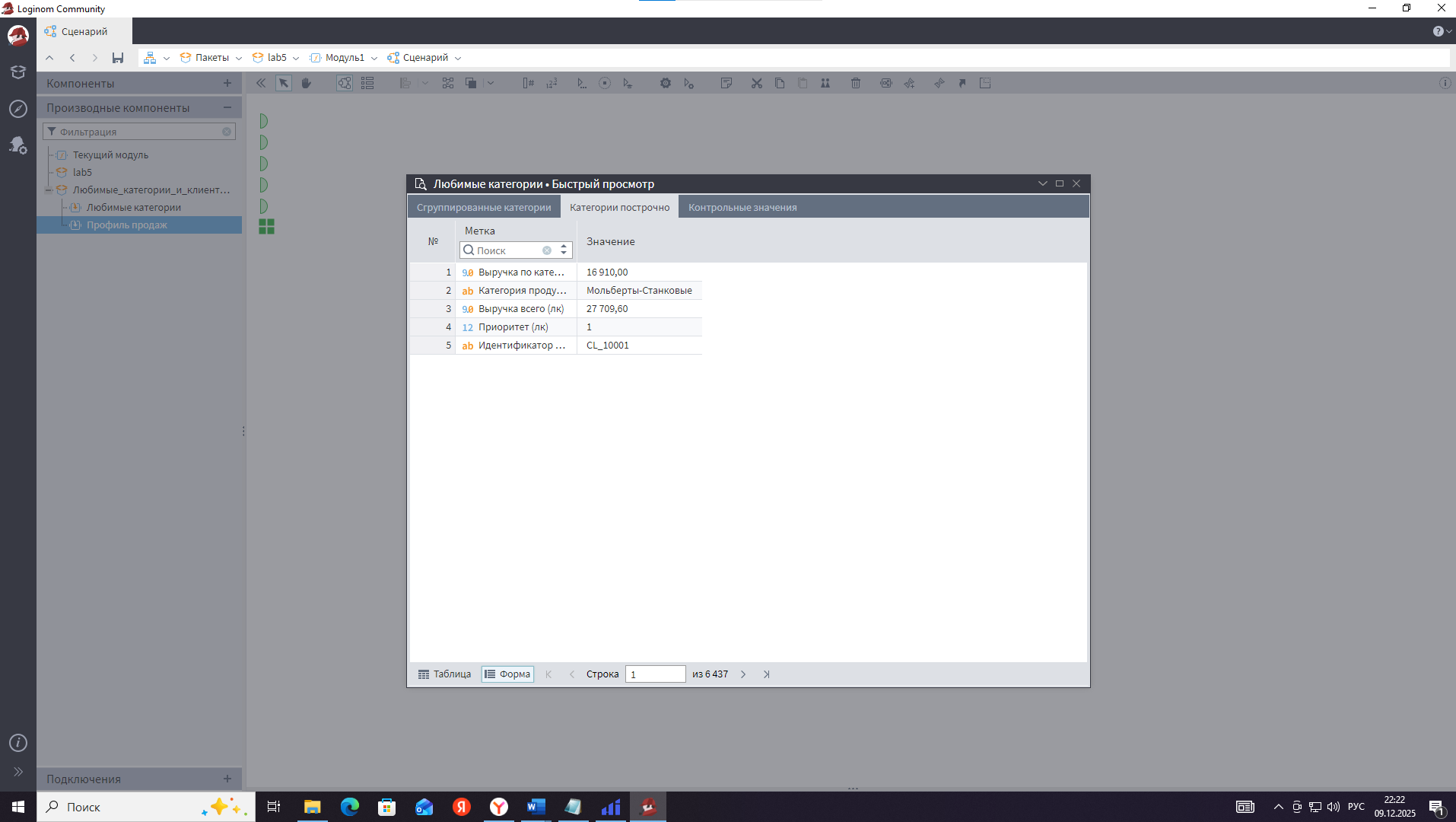
Проверяю оба варианта. В свернутой таблице вижу колонки: Client\_ID и, например, Favorite\_categories (строка категорий) – теперь у каждого клиента указаны 1-2-3 категории, которые занимают наибольшую долю в его покупках

**

**

В развернутой форме – множество строк, где каждый клиент может встречаться несколько раз (если у него несколько любимых категорий), с указанием категории и доли, которую она занимает

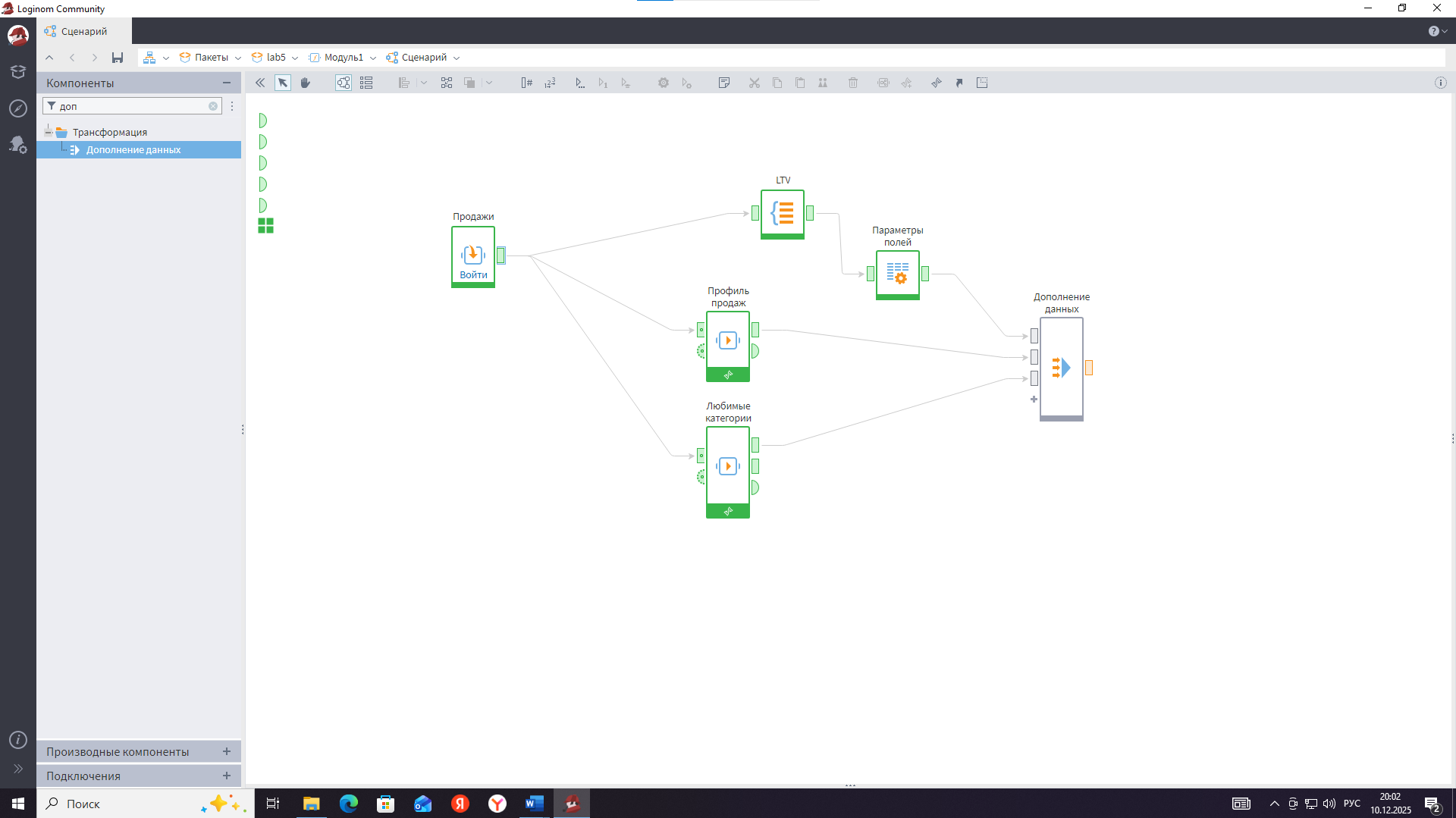
**

**

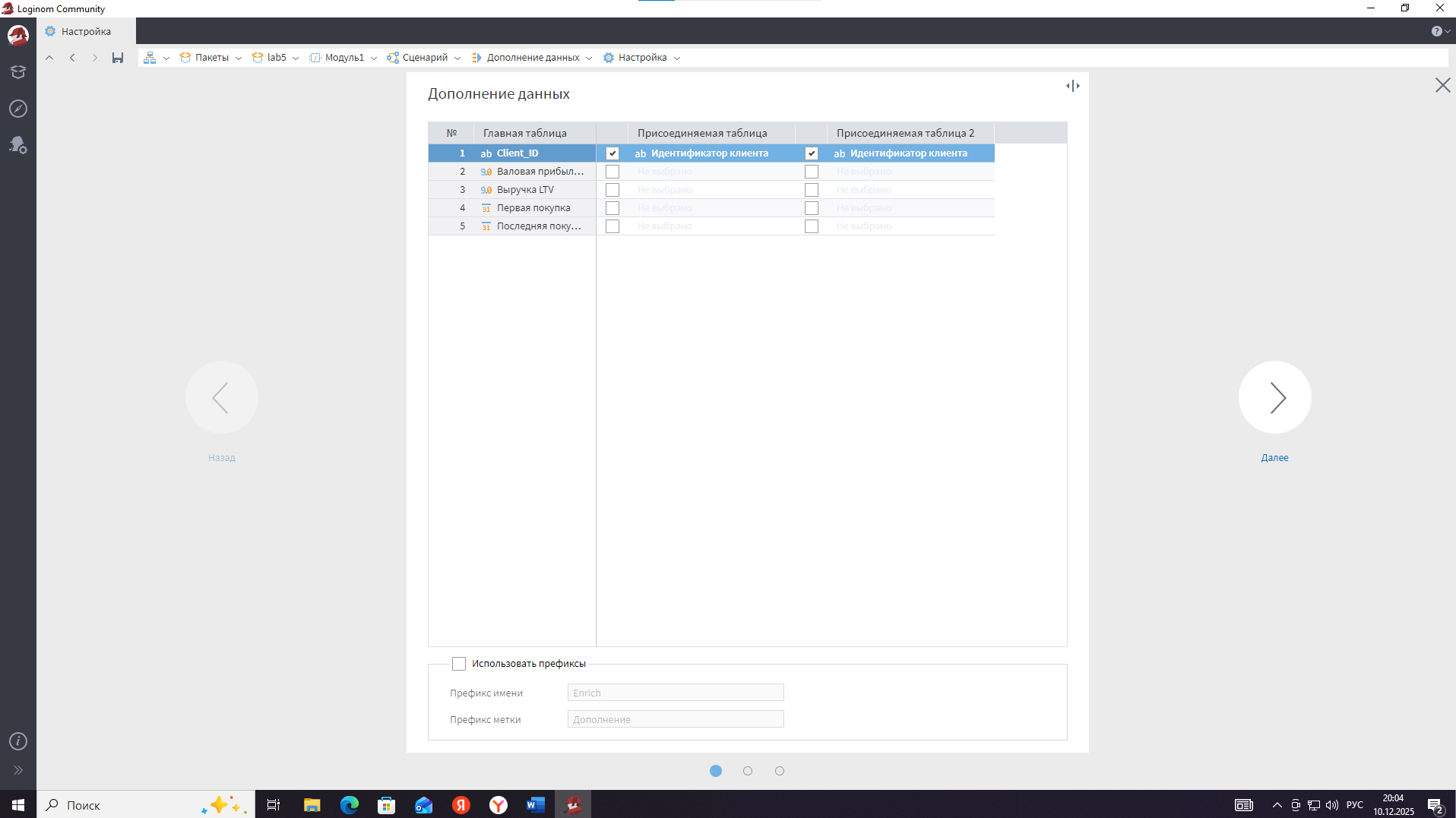
Эти данные пригодятся для дальнейшего анализа или фильтрации.

1. Объединение данных в единый справочник клиентов (узел «Дополнение данных»). На данном этапе у нас есть несколько таблиц с разной информацией на уровне клиента:
2. Базовая таблица LTV (Client\_ID, LTV-выручка, LTV-прибыль, первая/последняя покупка).
3. Таблица профиля продаж (Client\_ID и показатели частоты/среднего чека/выручки за период и т.д.).
4. Таблица любимых категорий (Client\_ID и список категорий).
5. (Опционально может быть еще исходный справочник клиентов, если он содержит какую-то дополнительную инфу, но в данном случае достаточно имеющихся).

Задача – соединить все эти данные в одну «широкую» таблицу, чтобы для каждого клиента в одной строке были все показатели. Использую узел «Дополнение данных» (в англ. Merge или Append by keys). Размещаю его на схеме. Подключаю: - Основную (верхнюю) таблицу: пусть это будет таблица LTV (она у нас уже содержит всех клиентов, 3184 строки). - 2-й вход: таблица, выходящая из компонента Профиль продаж. - 3-й вход: таблица (свернутая) из компонента Любимые категории. - 4-й вход: при желании можно добавить исходный справочник клиентов (чтобы, например, включить названия организаций или другие данные, если нужны).

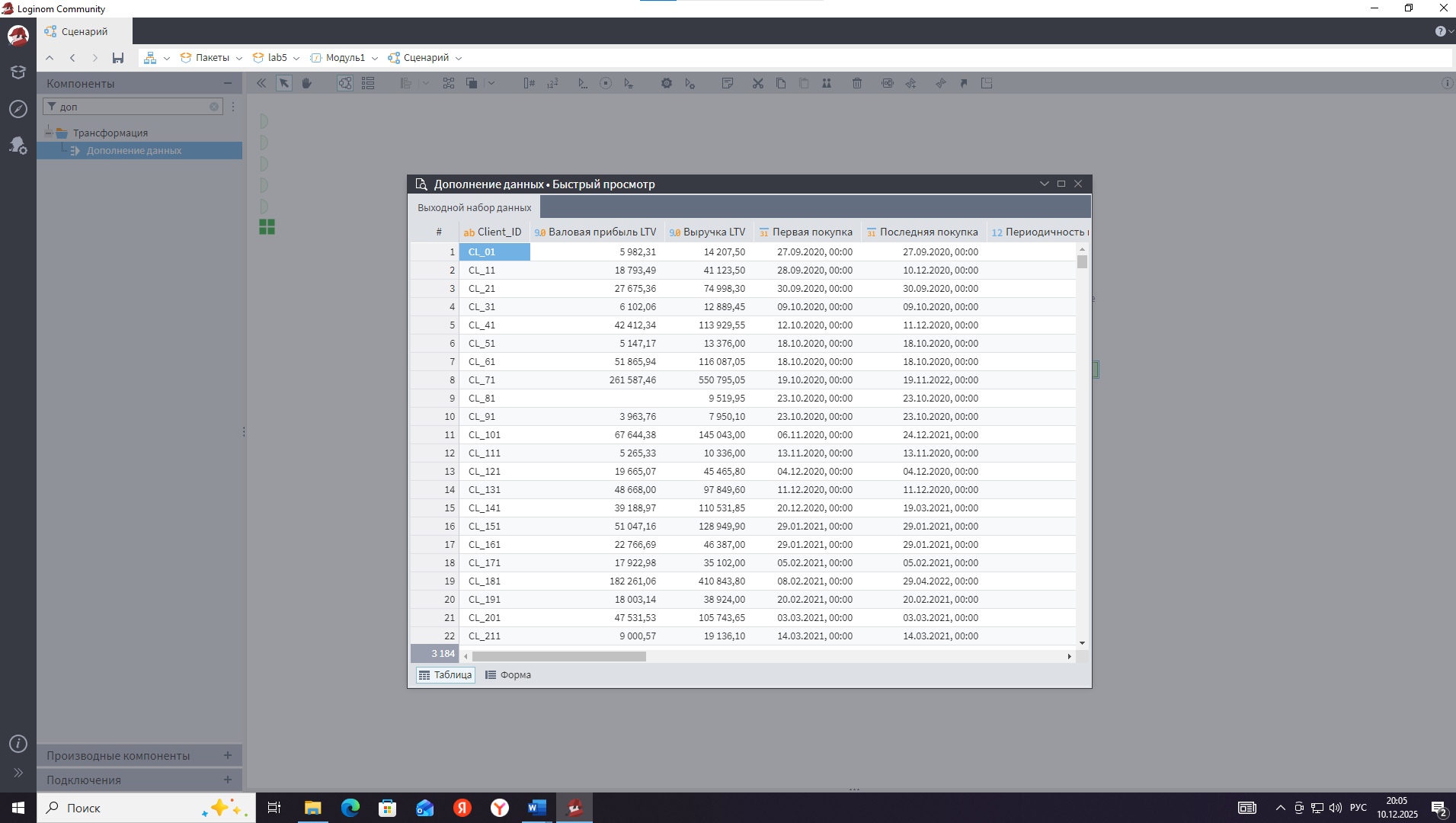
**

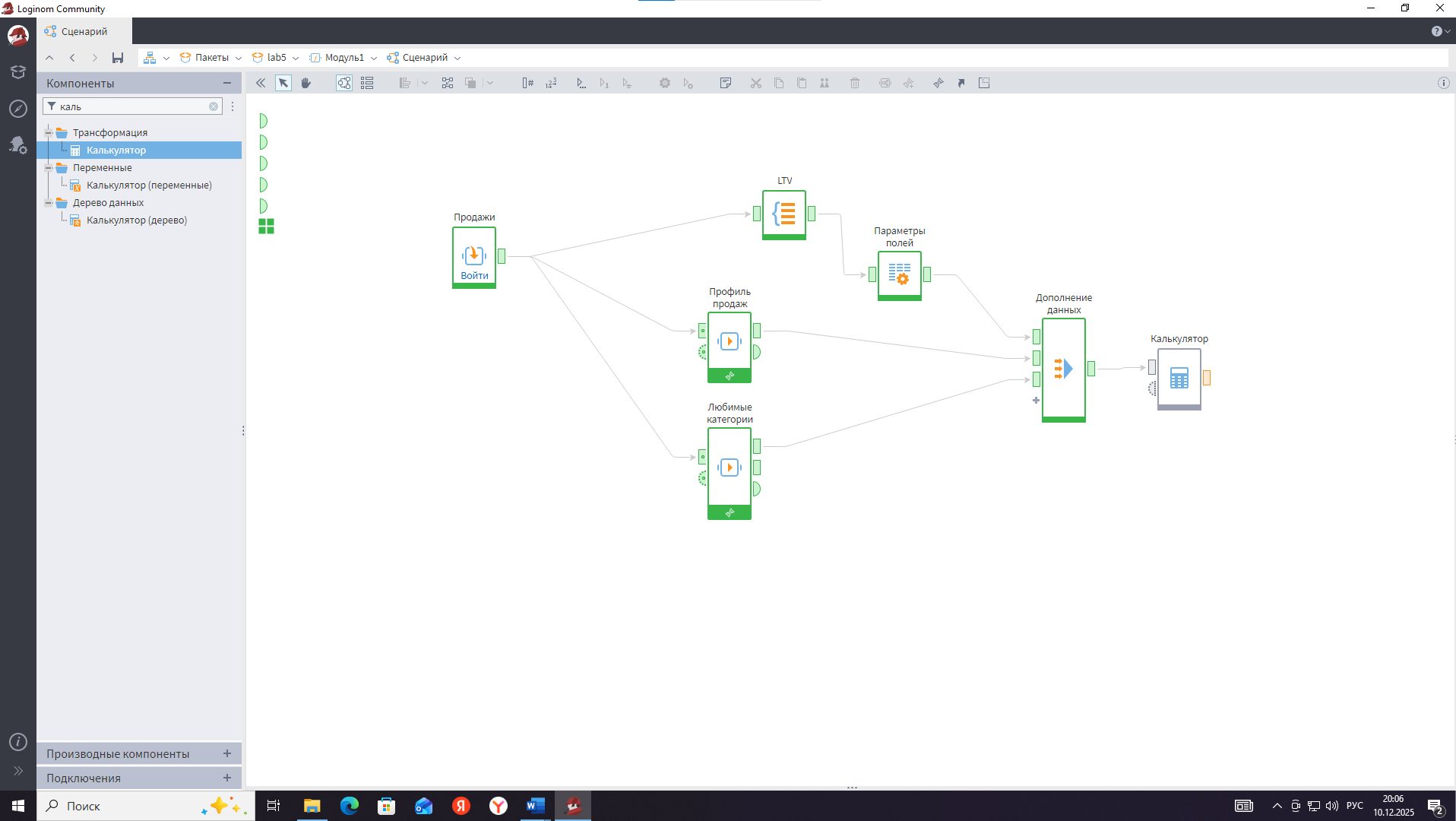
Открываю настройки узла «Дополнение данных». В мастере, на первом шаге, указываю ключ для объединения – Client\_ID для главной таблицы. Далее сопоставляю его с полями во всех присоединяемых таблицах (как правило, они тоже называются Client\_ID). Выбираю, какие поля из присоединяемых таблиц добавить: можно включить все, либо выборочно. В нашем случае беру все рассчитанные показатели профиля и любимых категорий, так как они все полезны

**

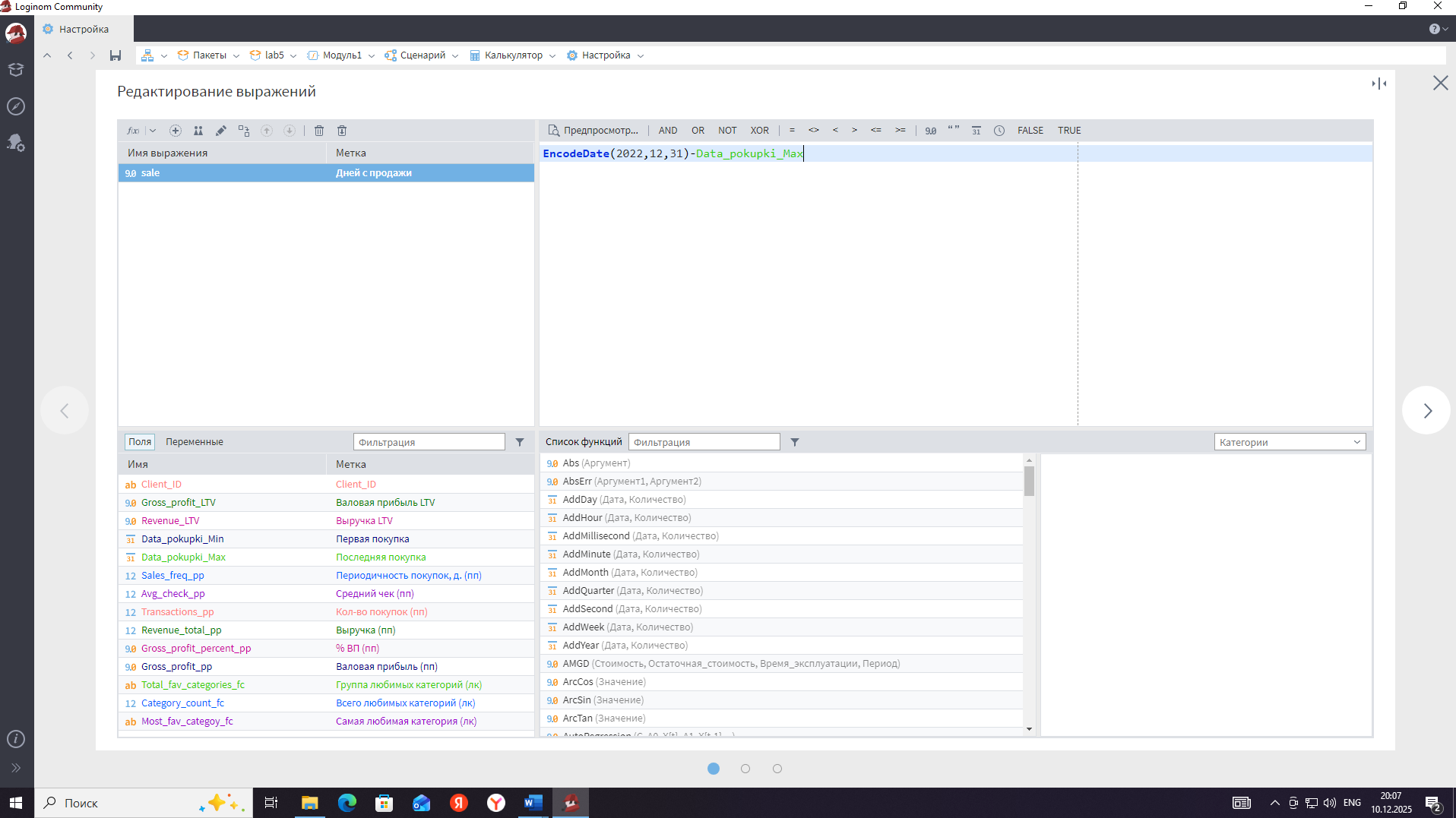
Сохраняю настройки и выполняю узел.

Результат – на выходе узла Дополнение данных получаем единый справочник клиентов, где каждая строка соответствует одному клиенту и содержит: - ID клиента, суммарную выручку/прибыль за все время, дату первой и последней покупки (из LTV); - Число покупок за последний год, средний чек, среднюю прибыль, периодичность и т.п. (из профиля продаж); - Любимые категории (например, текстовое поле со списком) из компонента категорий; - Возможно, информацию из исходного справочника (название юрлица и пр., если добавляли).

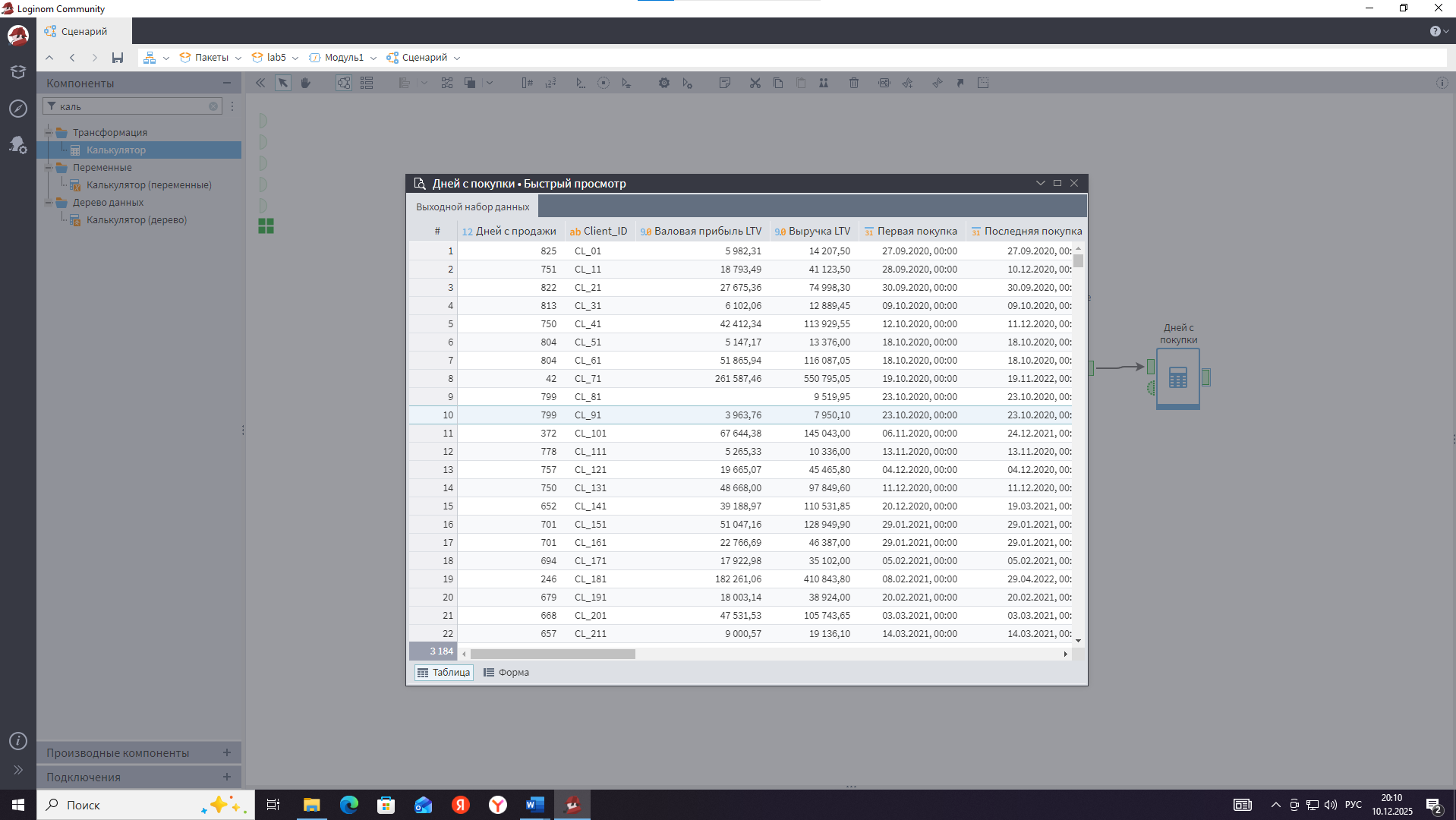
Число строк осталось 3184 (клиентов не убавилось и не прибавилось), а количество столбцов значительно увеличилось – объединились все данные **Расчет дополнительных признаков (Калькулятор). Теперь, имея полный набор информации по каждому клиенту, можно вывести несколько категориальных признаков для сегментации: статус, уровень, активность и т.д. Добавляю узел «Калькулятор» и связываю его с выходом справочника клиентов (выход Дополнение данных -> вход Калькулятор). Открываю мастер настроек.



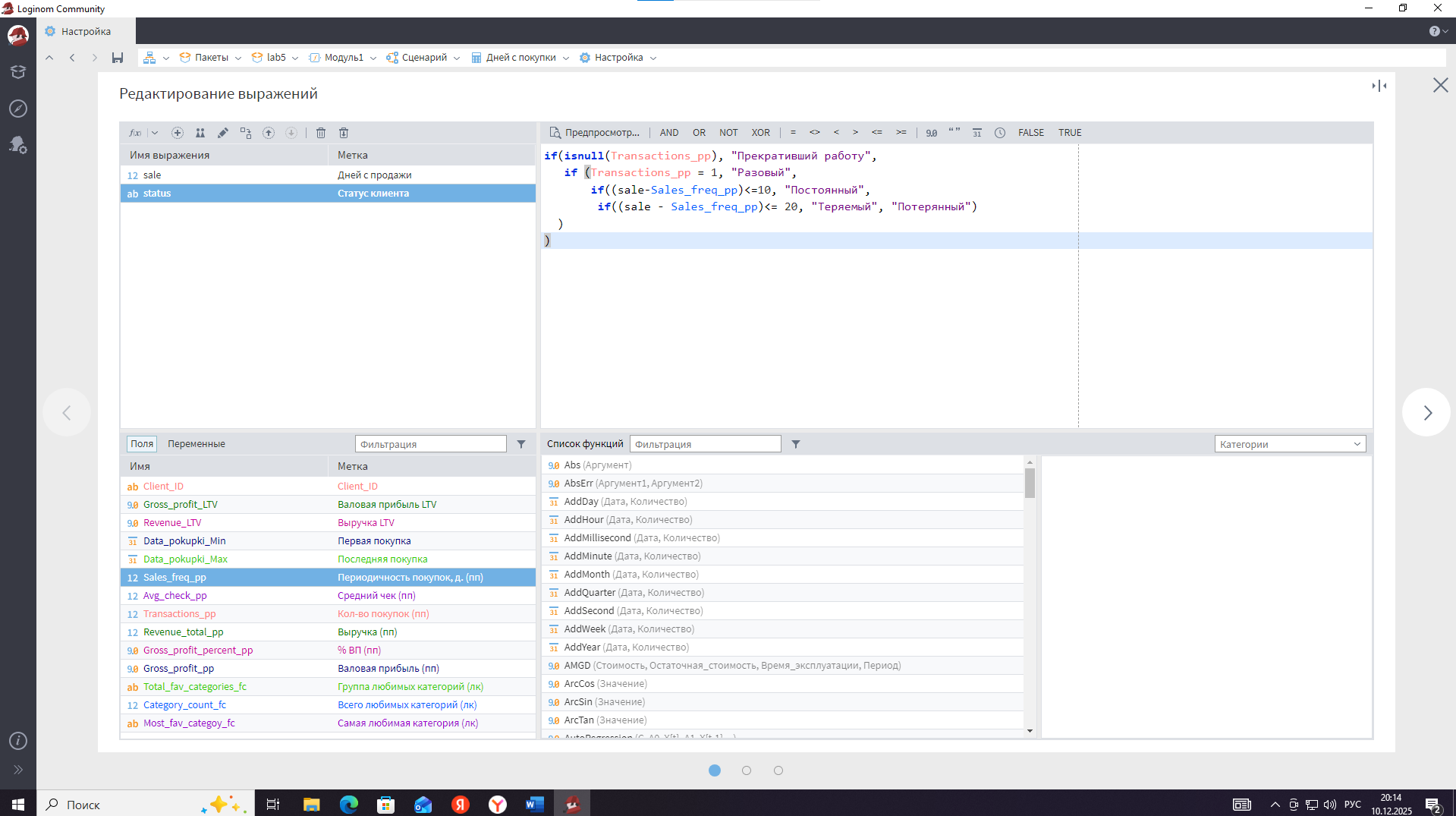
Создаю несколько вычисляемых полей: - Days\_from\_sale (Дней с последней покупки). Тип данных – целое число. Формула: беру текущую дату (функция today() или последний доступный день) и вычитаю поле *Last\_purchase\_date*. Это даст количество дней между последней покупкой клиента и сегодня. Формат записи: today() - Last\_purchase\_date (Loginom определит, что разность дат нужно выразить в днях)



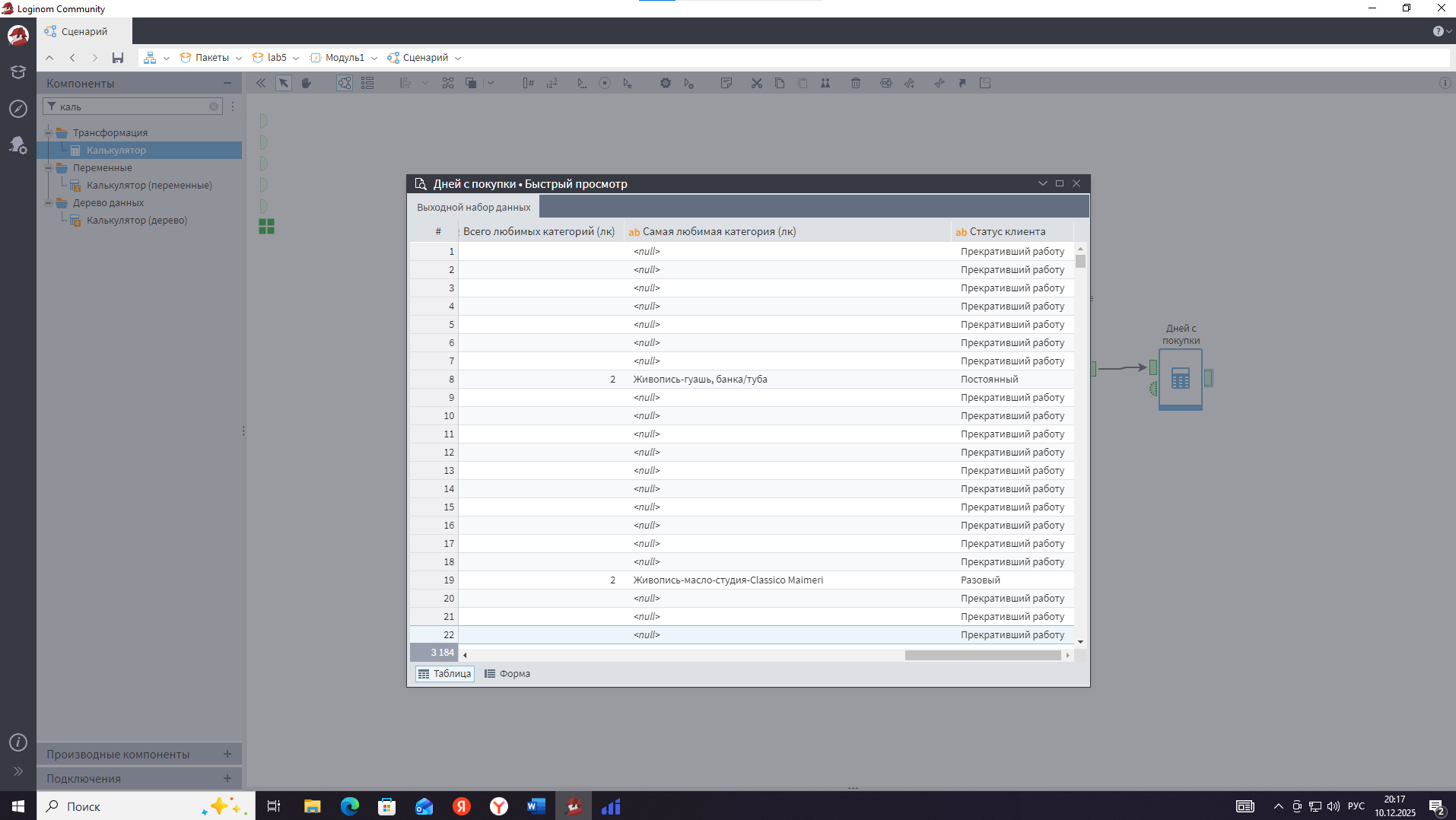
Сохраняю. Проверяю: у активных клиентов (кто недавно покупал) значение мало, у давно неактивных – большое число.



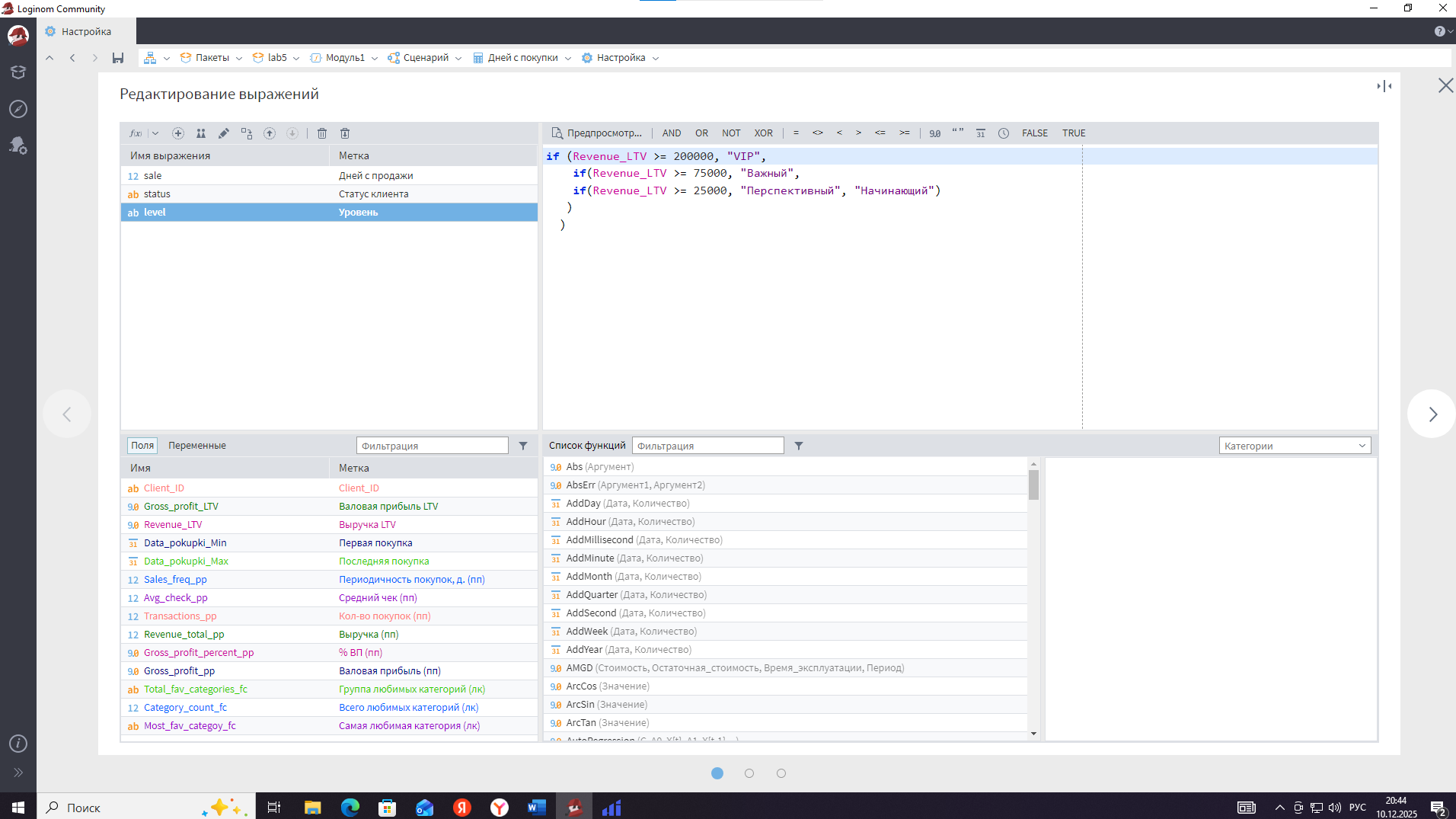
* Client\_status (Статус клиента). Тип – текст. Ввожу условие, которое разделит клиентов на категории по значимости. Например, можно присвоить статус «VIP клиент» тем, у кого LTV выручка выше определенного порога, или у кого за последний год сумма покупок высокая, или более X покупок. Для примера сделаю так: если количество покупок (поле *Purchases\_count*, рассчитанное в профиле) больше 10, считаем клиента VIP, иначе – обычный. Формула: if(Purchases\_count > 10, "VIP клиент", "Обычный клиент")



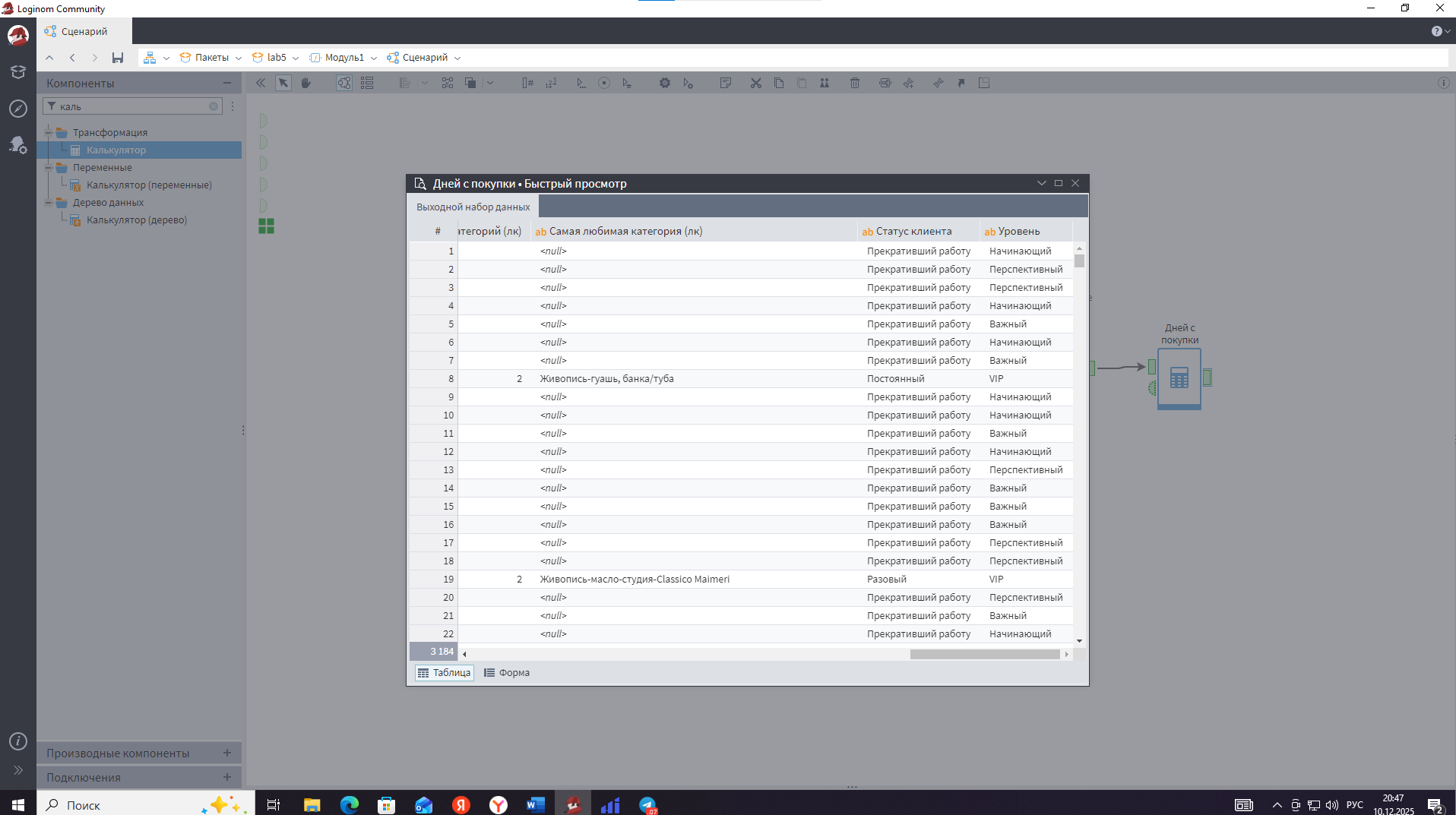
* После выполнения поля Client\_status во всех строках заполнится текстом «VIP клиент» или «Обычный клиент». Например, 5% самых активных могут получить VIP, остальные – обычные



* Client\_level (Уровень клиента). Тоже текстовый признак, но уже на основе суммарной выручки LTV (всей истории). Ввожу условие с несколькими градациями. Например: если LTV выручка > 100000 – присваиваю «Премиум»; иначе если > 50000 – «Золотой»; иначе – «Стандарт». Это трехуровневая сегментация по денежному объему. Формула с вложенным IF может выглядеть так:
* if(Revenue\_LTV > 100000, "Премиум",  
   if(Revenue\_LTV > 50000, "Золотой", "Стандарт"))



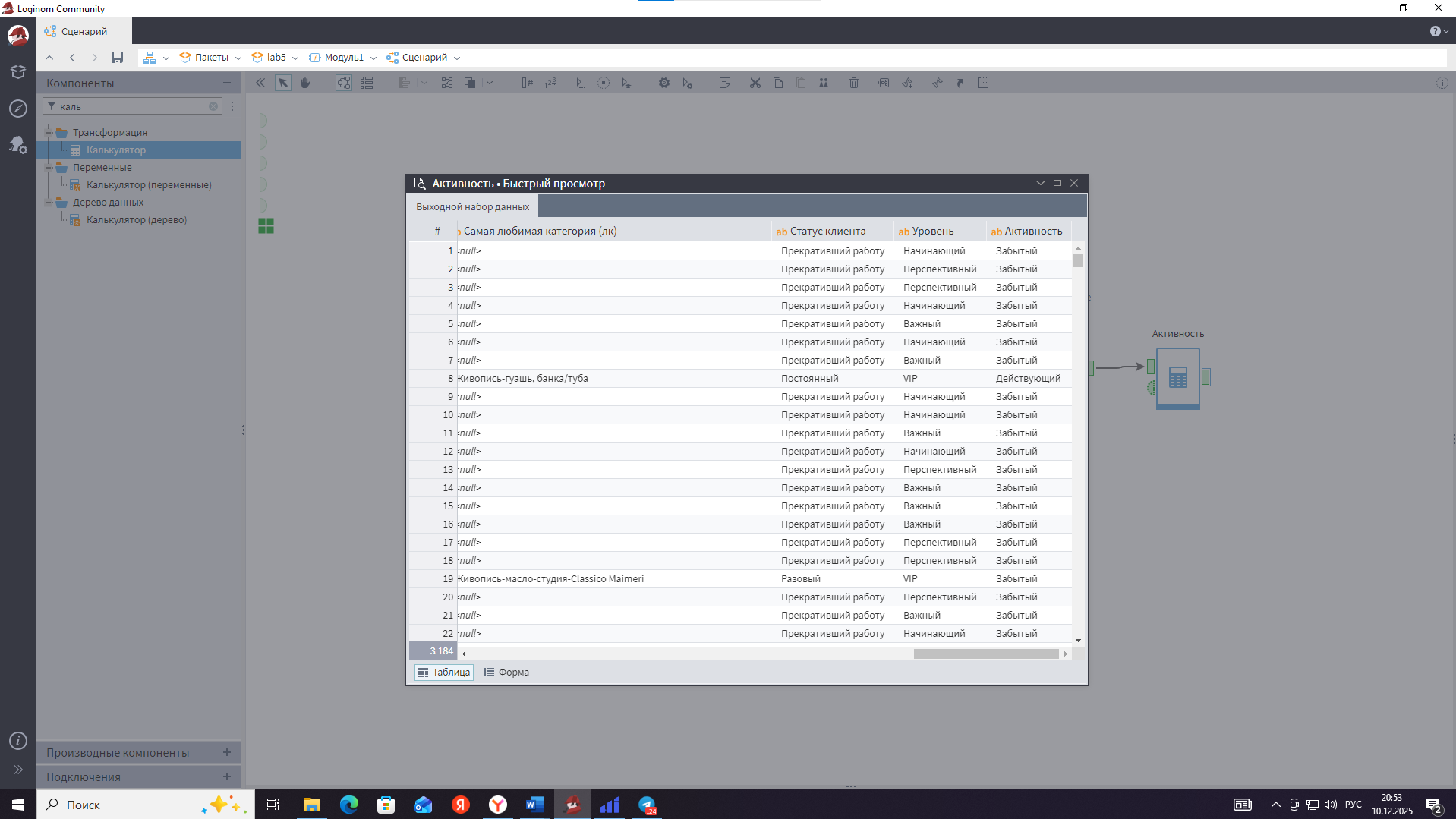
* После выполнения, каждому клиенту присваивается уровень, отражающий его ценность для компании



* Client\_activity (Активность клиента). Здесь будем использовать рассчитанный ранее Days\_from\_sale. Определим, считать ли клиента активным или нет. Например, правило: если с последней покупки прошло более 180 дней (полгода), то «Неактивный», иначе «Активный». Формула: if(Days\_from\_sale > 180, "Неактивный", "Активный")



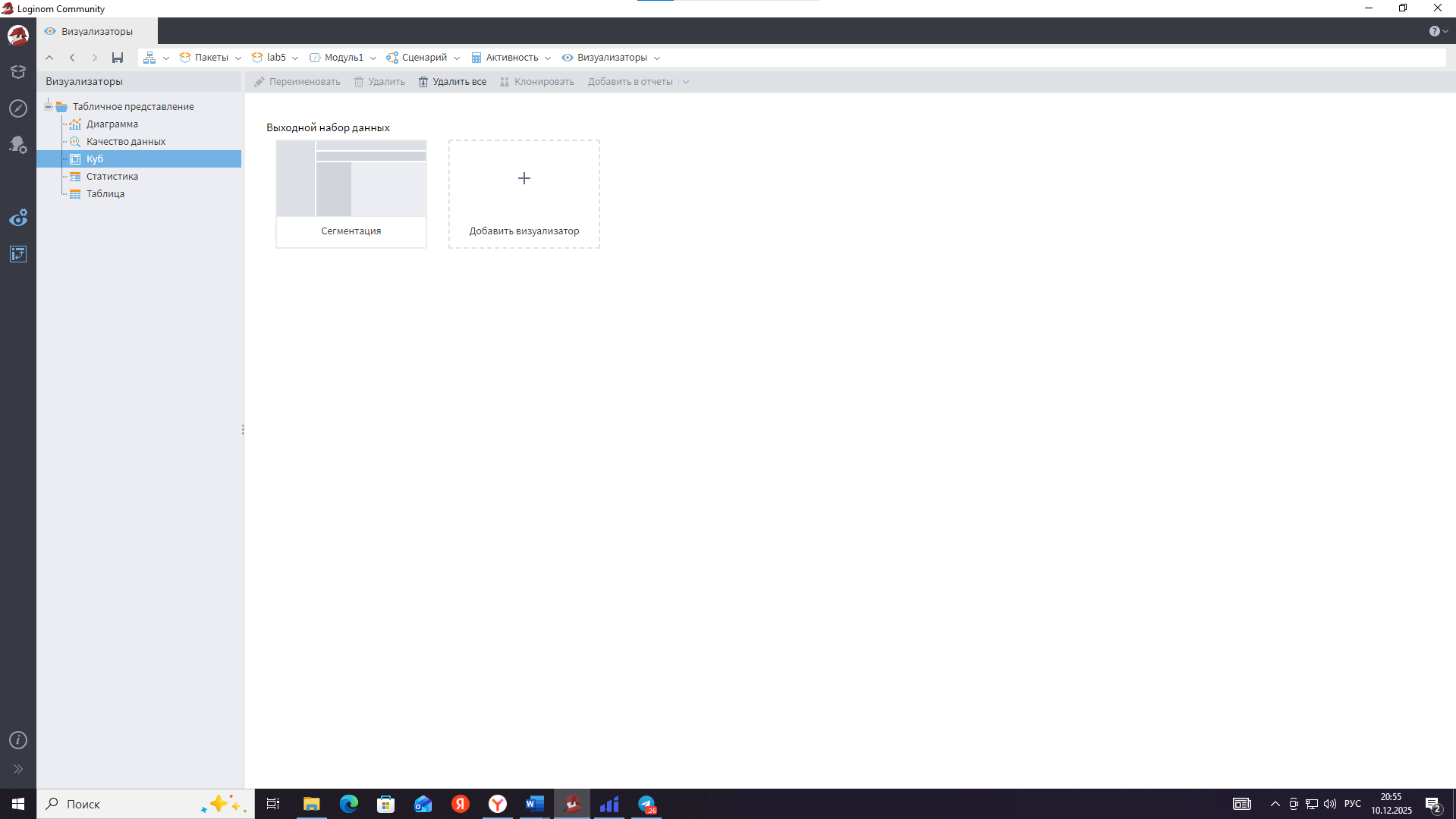
* Выполняю – появляется поле с меткой «Активность клиента», заполненное «Активный» или «Неактивный» для каждого клиента



При желании, можно вычислить и другие признаки (например, segment по любимым категориям или регион по адресам – но у нас таких данных может не быть). Закрываю настройки калькулятора и выполняю узел. Теперь итоговый справочник клиентов обогащен новыми полями, готовыми для построения сегментированных отчетов.

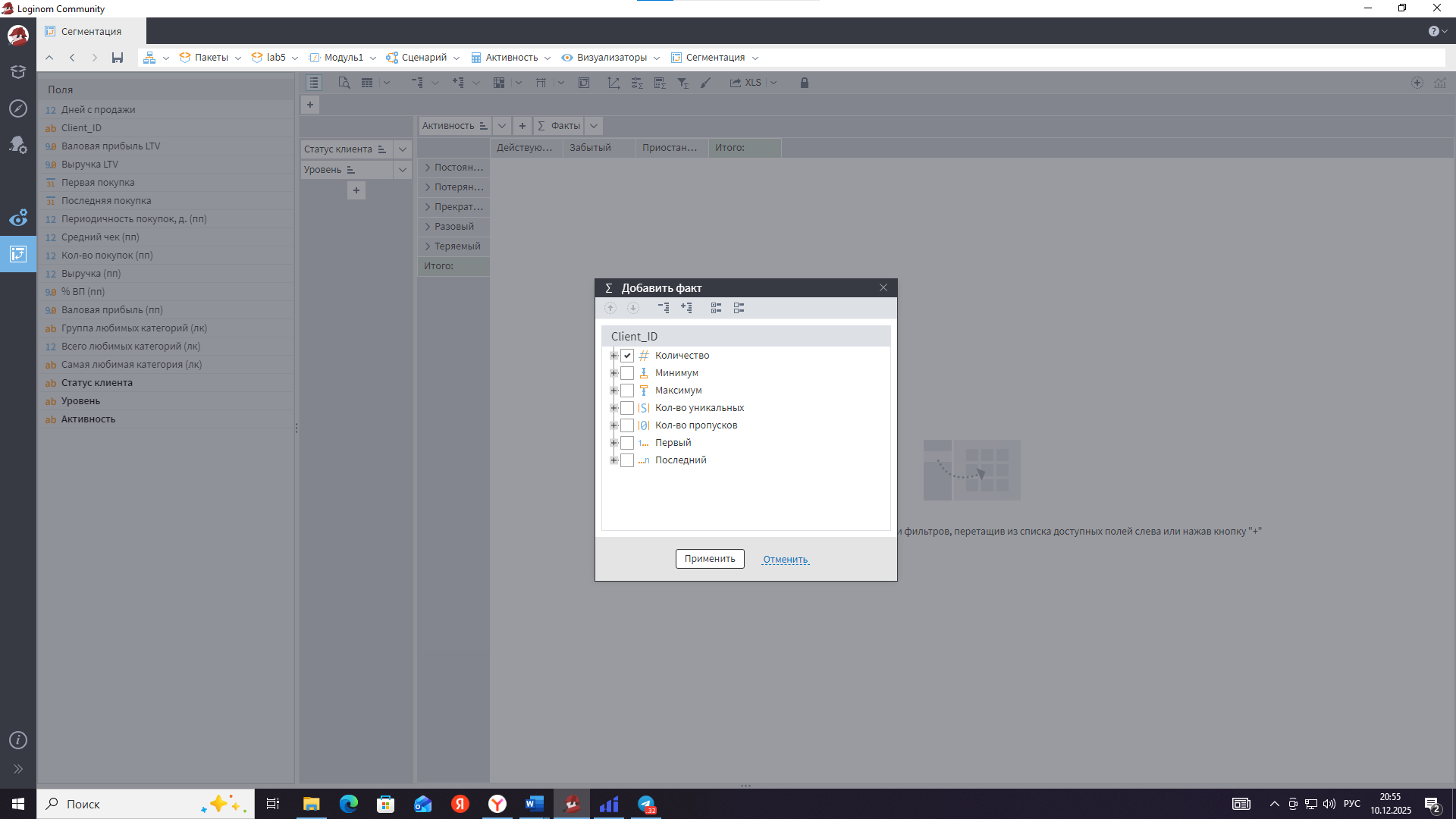
**Домашнее задание:**

Отчёт 1: Сегментация клиентов. Требуется построить сводный куб, показывающий распределение клиентов по сегментам (статус, уровень, активность). Использую финальный узел Калькулятор (у него есть все данные). Нажимаю значок глаза -> Добавить визуализатор -> Куб. Называю визуализатор, например, «Сегментация клиентов»

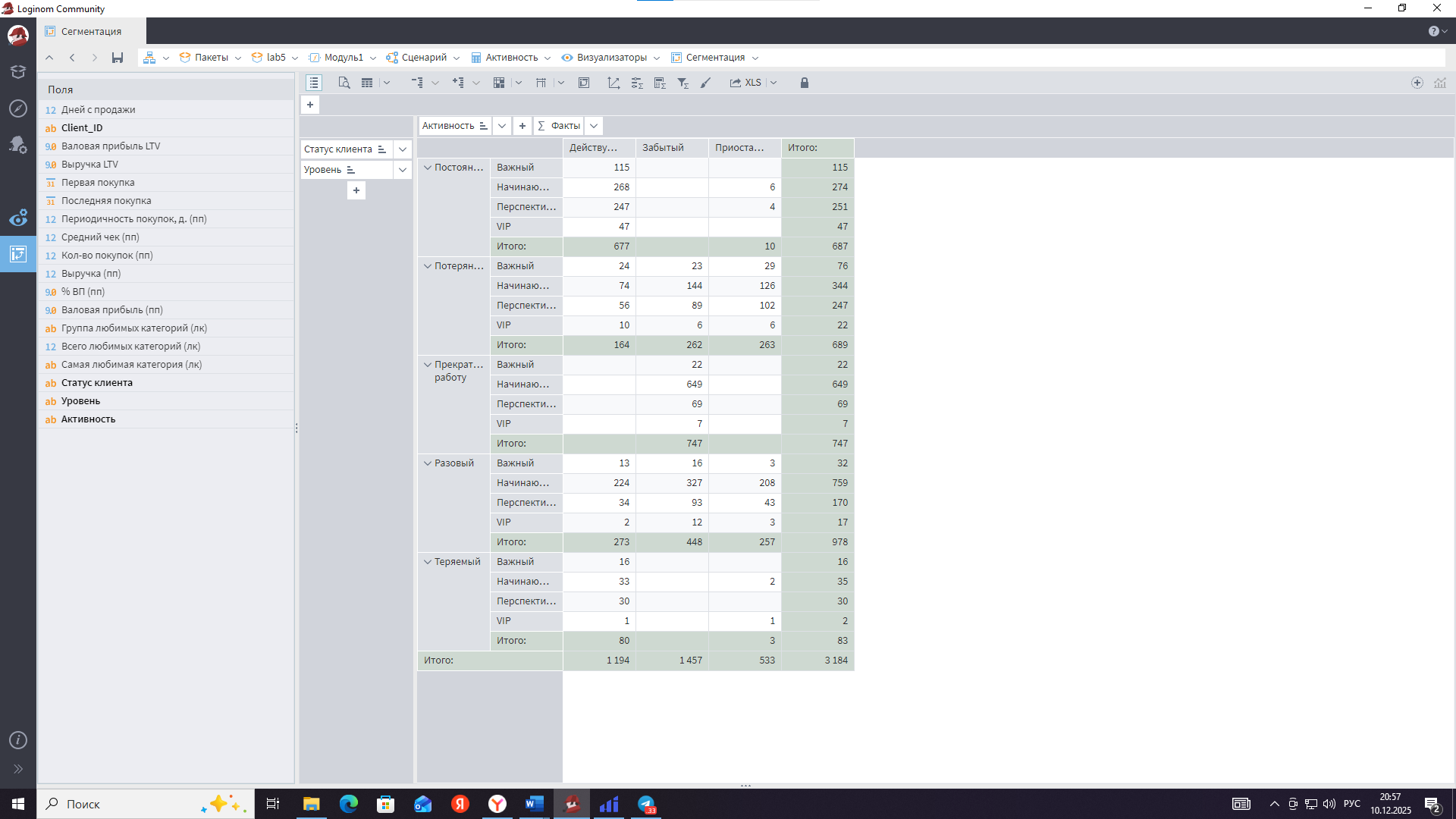
**

В конфигураторе куба задаю:

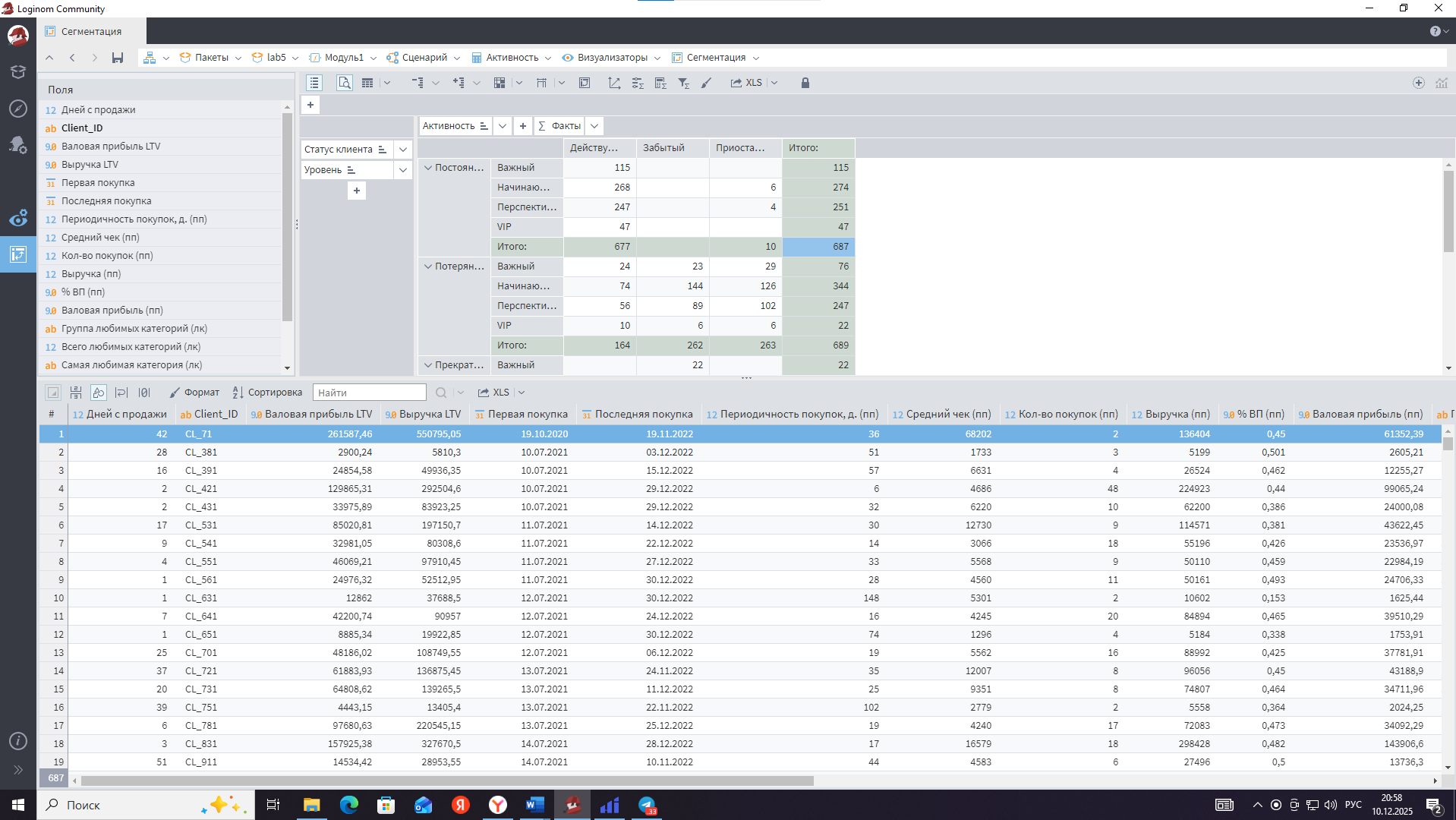
* Строки: перетаскиваю поле *Client\_status* (статус: VIP/Обычный), а ниже него – *Client\_level* (уровень: Премиум/Золотой/Стандарт). Таким образом, строки куба группируются сначала по статусу, внутри – по уровню.
* Столбцы: добавляю поле *Client\_activity* (активность: Активный/Неактивный). Это поделит данные на два столбца – активные и неактивные клиенты.
* Показатели: перетаскиваю поле *Client\_ID*, и в его настройках выбираю агрегирование «Количество (уникальное)». По сути, это подсчет количества клиентов в каждом сегменте



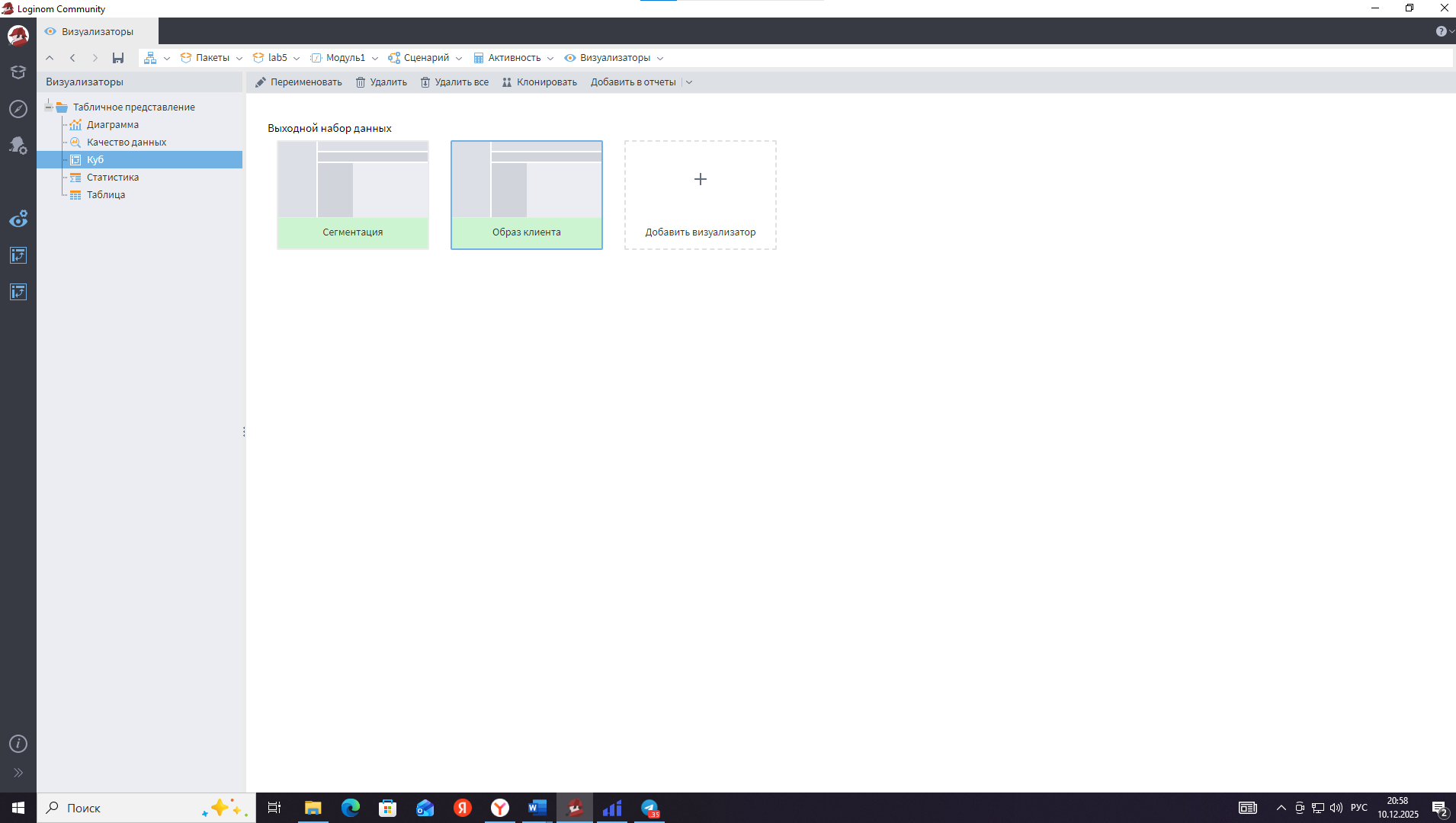
* Сохраняю визуализатор. Получается таблица, где для каждого сочетания (например, VIP-Премиум активные, VIP-Премиум неактивные, ... Обычный-Стандарт активные/неактивные) стоит число клиентов. Сразу видно, например, сколько у нас активных VIP клиентов и сколько «уснувших» стандартных клиентов и т.п.



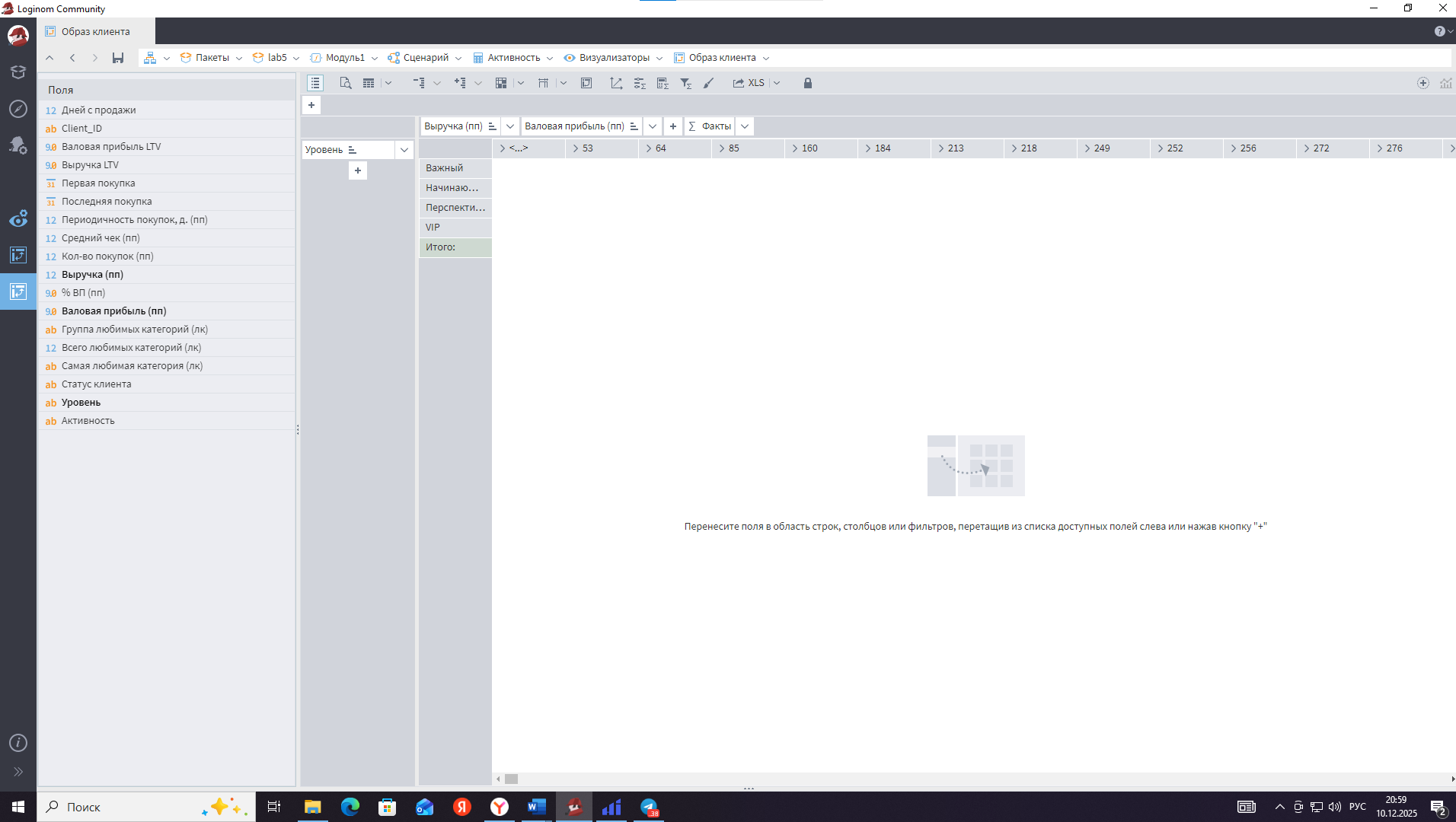
* Чтобы можно было изучить конкретных клиентов в сегментах, включаю детализацию. В настройках куба включаю галочку «Детализация», выбираю опцию показать список записей при проваливании. Теперь, если щелкнуть по числу в ячейке, снизу откроется список конкретных Client\_ID (или даже имен, если бы они были) тех клиентов, которые попали в выбранный сегмент



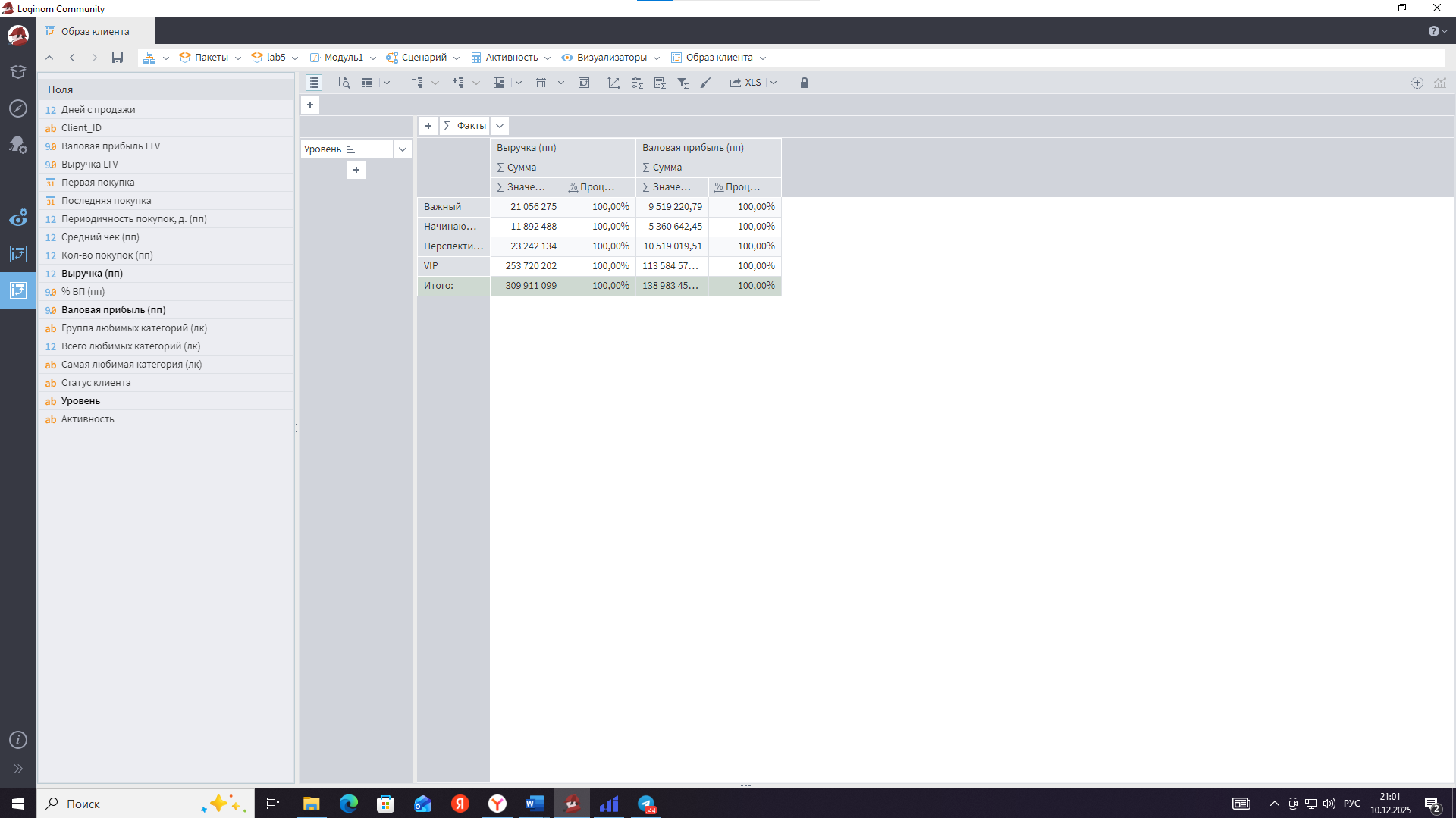
* Отчёт 2: Портрет клиента (финансовая привлекательность). Второй отчет будет показывать суммарные показатели по сегментам клиентов и средние значения. Опять же, на узле Калькулятор добавляю визуализатор «Куб», называю его «Образ клиента»



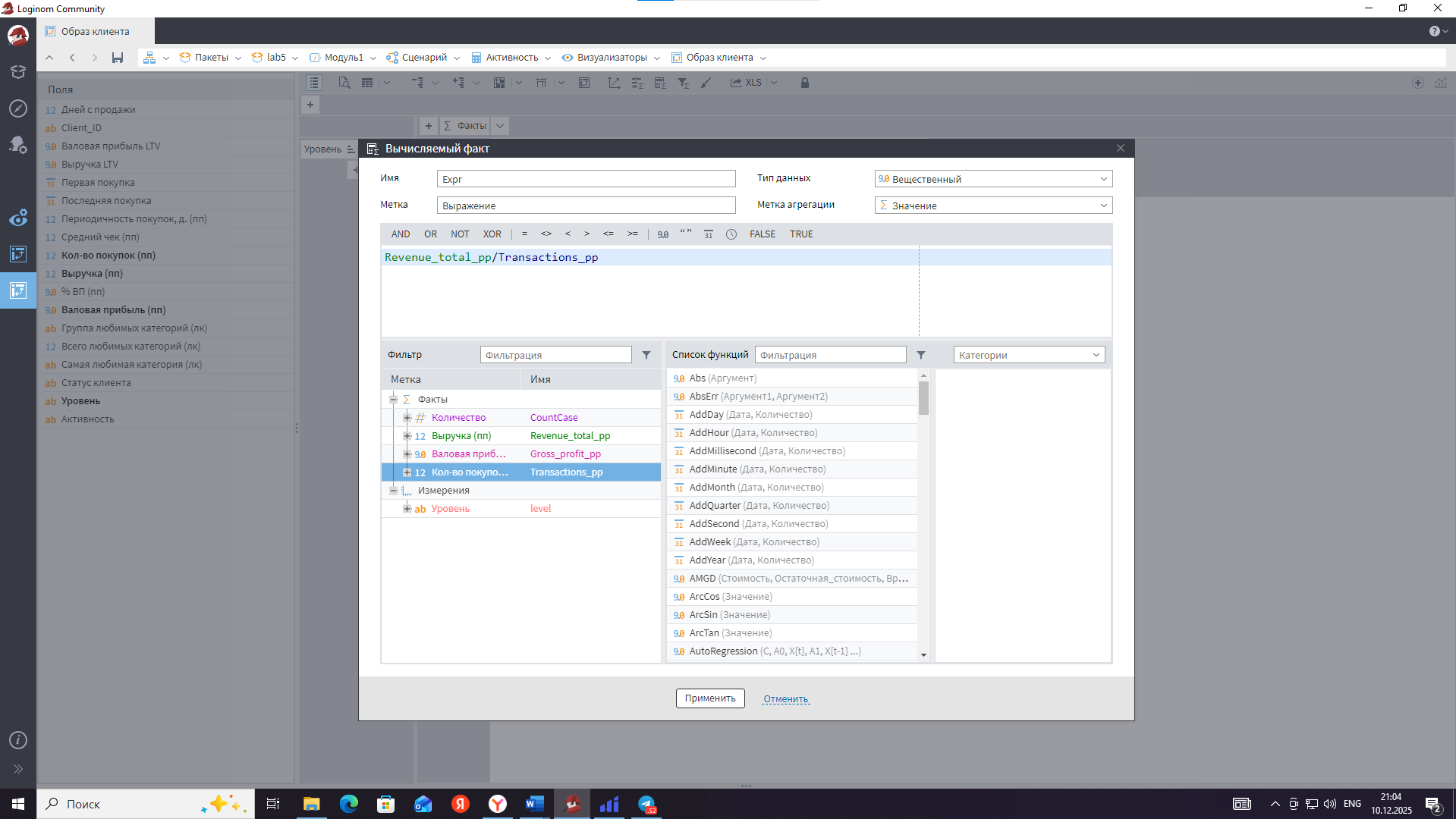
* Настраиваю следующим образом:
* Строки: поле *Client\_level* (уровень клиента).
* Столбцы: поле *Client\_status* (статус клиента).
* Показатели: добавляю три поля – *Revenue\_LTV* (суммарная выручка от сегмента клиентов), *Gross\_profit\_LTV* (суммарная валовая прибыль сегмента) и *Purchases\_count* (общее число покупок, совершенных сегментом). Для всех указываю агрегирование Сумма, чтобы получили итоги по группе



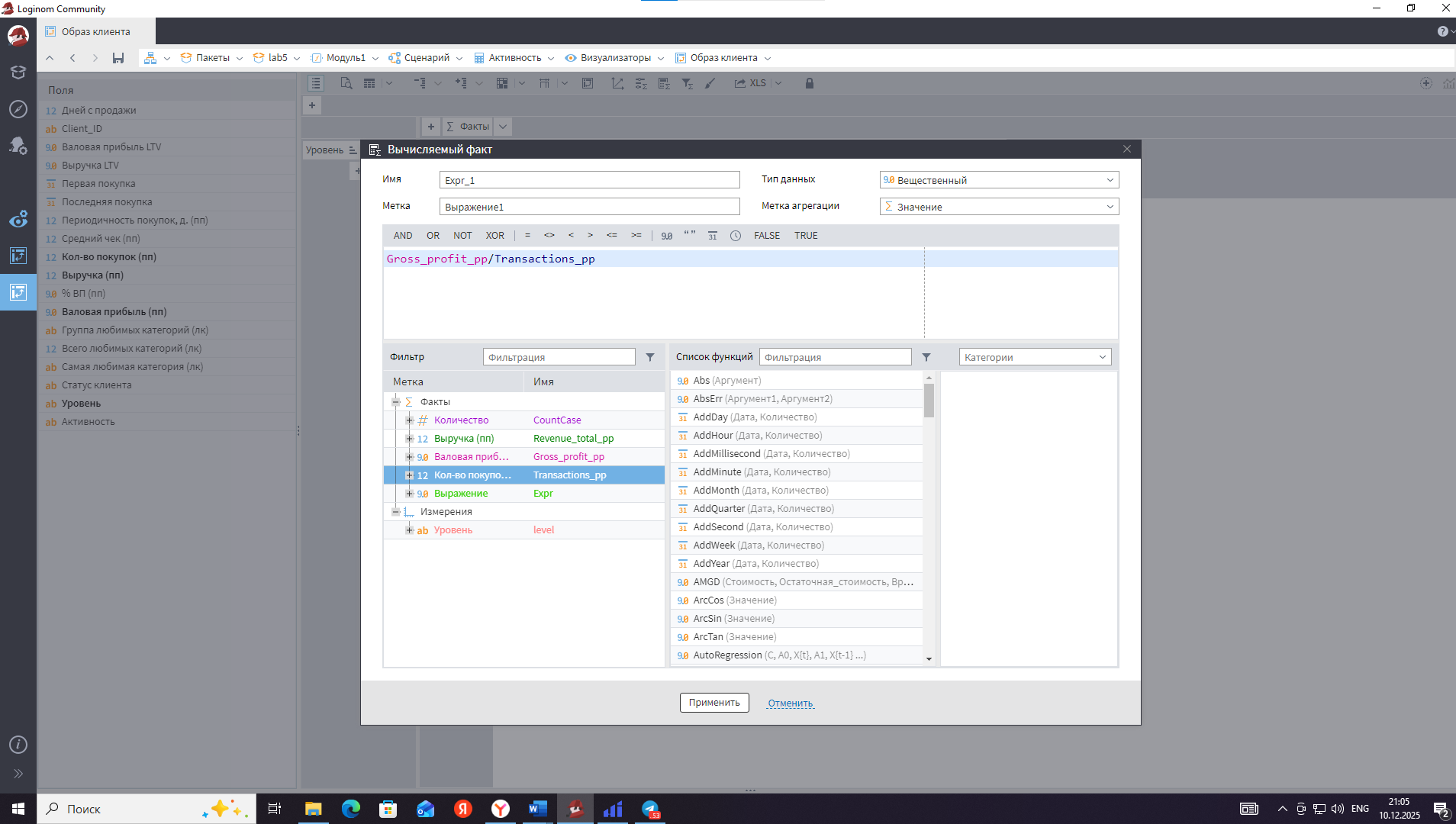
* Затем добавляю несколько производных метрик: - Доля выручки сегмента: выделяю показатель Revenue\_LTV и выбираю опцию «Добавить факт -> % по горизонтали» (то есть процент от суммы по строке). Аналогично делаю для Gross\_profit\_LTV и Purchases\_count – добавляю для каждого долю по горизонтали



* После этого в отчете появятся дополнительные колонки вида «Выручка (%)», «Прибыль (%)», «Покупки (%)», показывающие какой процент от общей выручки всех статусов составляет данный статус в рамках уровня, и т.п. - Средний чек: рассчитываю как выручка / количество покупок. Для этого жму «Добавить вычисляемый факт», задаю имя Avg\_check и метку «Средний чек». В формуле указываю Revenue\_LTV / Purchases\_count

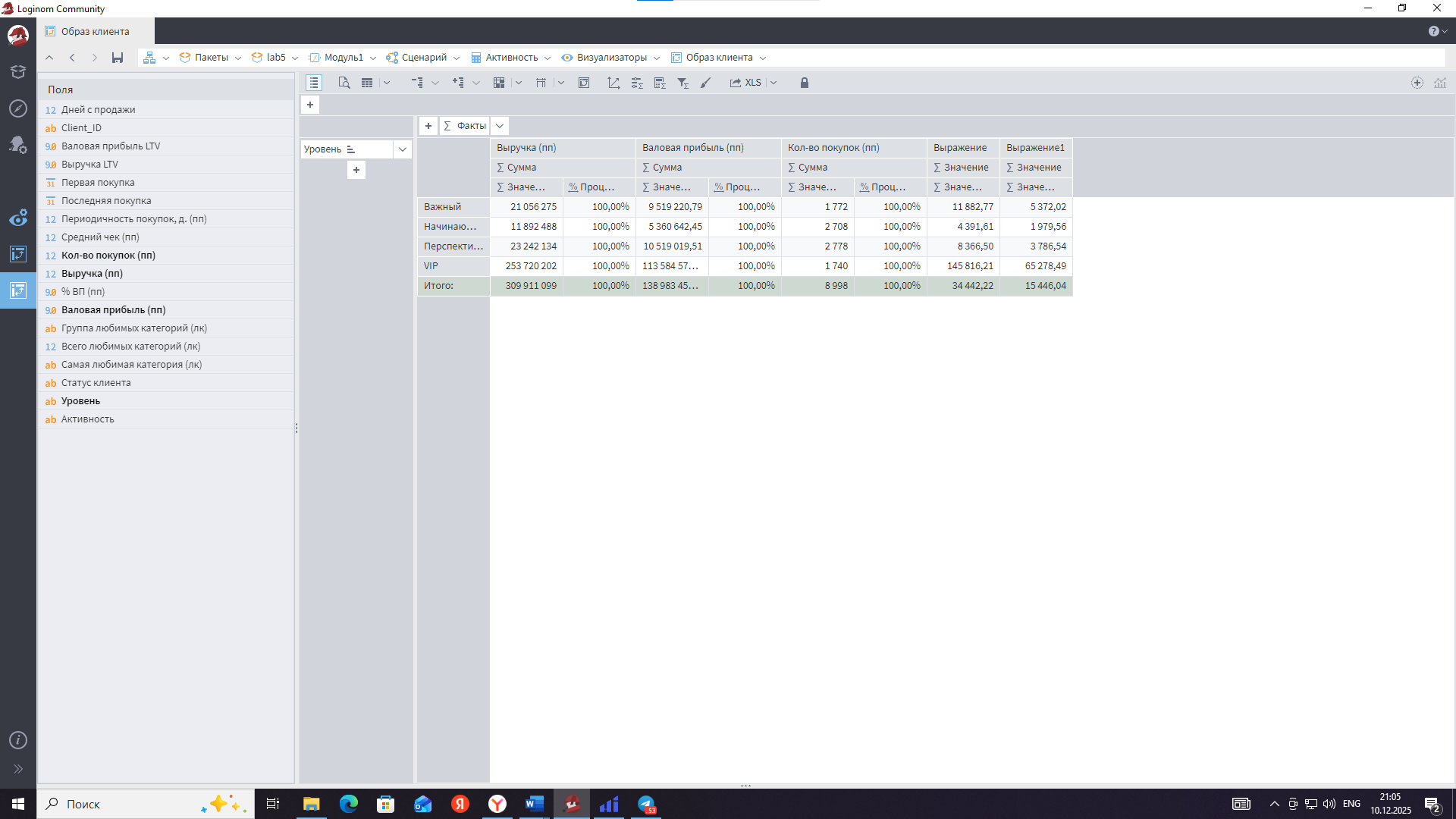


* - Средняя прибыль на покупку: аналогично, добавляю факт Avg\_profit (метка «Средняя прибыль»), формула Gross\_profit\_LTV / Purchases\_count



Сохраняю визуализатор. Теперь отчет «Портрет клиента» показывает для каждой комбинации уровня и статуса: - Общее число клиентов, - Суммарную выручку и прибыль, а также их долю относительно других статусов (внутри уровня), - Общее число покупок, - Средний чек и среднюю прибыль на одну покупку.

На рисунке представлен пример такого куба: можно видеть, например, сколько в сумме принесли «Премиум»-клиенты, какой у них средний чек, отличается ли он у VIP и обычных, и т.д.



**Выводы**

Лабораторная работа была посвящена углубленному анализу клиентской базы. В результате выполнения работы были получены следующие результаты и закреплены навыки: - Мы научились структурировать сложный сценарий, группируя узлы в подмодели. Это позволило сделать схему более понятной и переиспользуемой. - Были рассчитаны ключевые показатели на уровне клиента: общая выручка и прибыль (LTV), число покупок, средний чек, периодичность, последняя активность и др. Это дало нам количественную основу для сегментации. - Освоили применение готовых аналитических компонентов Loginom: «Профиль продаж» помог автоматически вычислить множество метрик (средние, частоты) без ручного кодирования, а «Любимые категории» выделил предпочтения клиентов. Мы научились подключать эти компоненты и правильно сопоставлять поля. - Скомбинировали результаты из разных источников в одном справочнике с помощью узла «Дополнение данных». Этот шаг наглядно показал прием объединения таблиц по ключу (аналог JOIN по Client\_ID), когда нужно совместить вычисления, выполненные разными ветками сценария. - С помощью «Калькулятора» создали дополнительные качественные признаки для клиентов, что позволило сегментировать их по уровню ценности, статусу, актуальности. Здесь были применены условные логические выражения для присвоения категорий на основе числовых показателей. - Построены наглядные многомерные отчеты (кубы) по клиентам: «Сегментация клиентов» демонстрирует структуру клиентской базы по нескольким критериям (статус/уровень/активность) с возможностью детализации до списка конкретных клиентов, а «Портрет клиента» дает сводную финансовую характеристику этих сегментов (сколько приносят, сколько их, каковы средние показатели).

Проанализировав отчеты, можно сделать практические выводы: например, какая категория клиентов наиболее ценна (приносит большую долю выручки), какова доля неактивных клиентов среди VIP и нужно ли проводить кампании по их возвращению, отличаются ли предпочтения товарных категорий у разных сегментов. Все это является ценнейшей информацией для маркетинговых решений и работы с клиентами.

В целом, выполнение этой лабораторной работы позволило существенно продвинуться от простого суммирования продаж до комплексной клиентской аналитики, что является важным шагом в построении системы управления взаимоотношениями с клиентами (CRM).