Департамент образования и науки города Москвы

Государственное автономное образовательное учреждение высшего образования города Москвы

«Московский городской педагогический университет»

Институт цифрового образования

Департамент информатики, управления и технологий

ДИСЦИПЛИНА:

Проектный практикум по разработке ETL-решений

**Самостоятельная работа 1**

**Интеграция данных из разных источников (баз данных)**

Выполнила: Сергеева А. И., группа: АДЭУ-211

Преподаватель: Босенко Т.М.

Москва

2025

**Цель работы:** разработка ETL-процесса для интеграции данных между PostgreSQL и MySQL с использованием Pentaho Data Integration.

**Задачи:**

- Создать исходные таблицы в PostgreSQL с различными наборами данных.

- Настроить целевые таблицы в MySQL для приема данных.

- Разработать процессы трансформации данных в Pentaho.

- Реализовать механизмы обработки ошибок и валидации данных.

- Создать представления для связанных данных.

**Необходимое ПО:**

- Конфигурация devops\_dba\_25.ova, можно скачать в разделе

http://95.131.149.21/moodle/mod/folder/view.php?id=1429

- учетная запись в Mysql.

**Ход работы:**

**Вариант st\_99. Работа с таблицами контрактов и их статусами, создание фильтра для активных заказов, рачет суммы и срока активности.** Для начала данные был выполнен пример из прошлой лабораторной. А именно выгрузка из CSV файлов, трансформации представлены на рисунках 1-3.

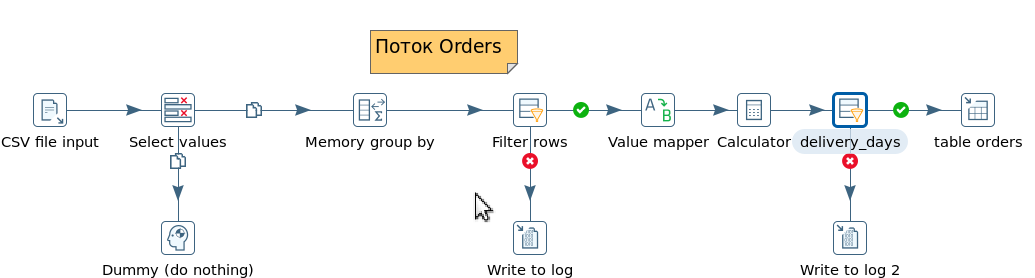


Рисунок 1 – Трансформация для таблицы orders

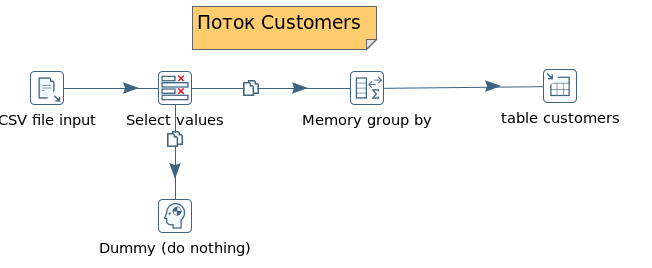


Рисунок 2 - Трансформация для таблицы customers

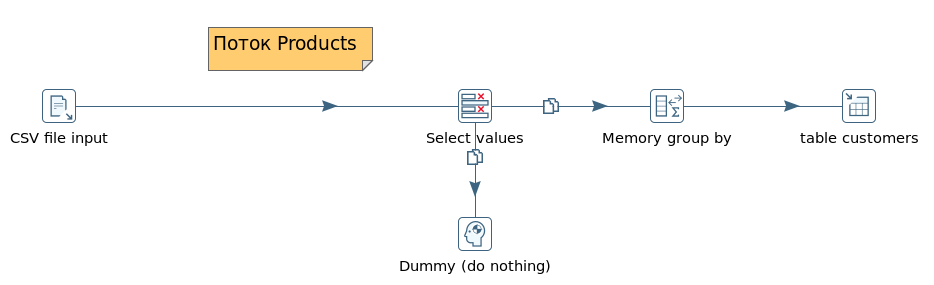


Рисунок 3 - Трансформация для таблицы

Следующим шагом было создание ещё одной трансформации для перекачки данных из PostgreSQL в MySQL. Для этого в рамках примера была создана в 2 базах таблица employees, как показано на рисунках 4-5.

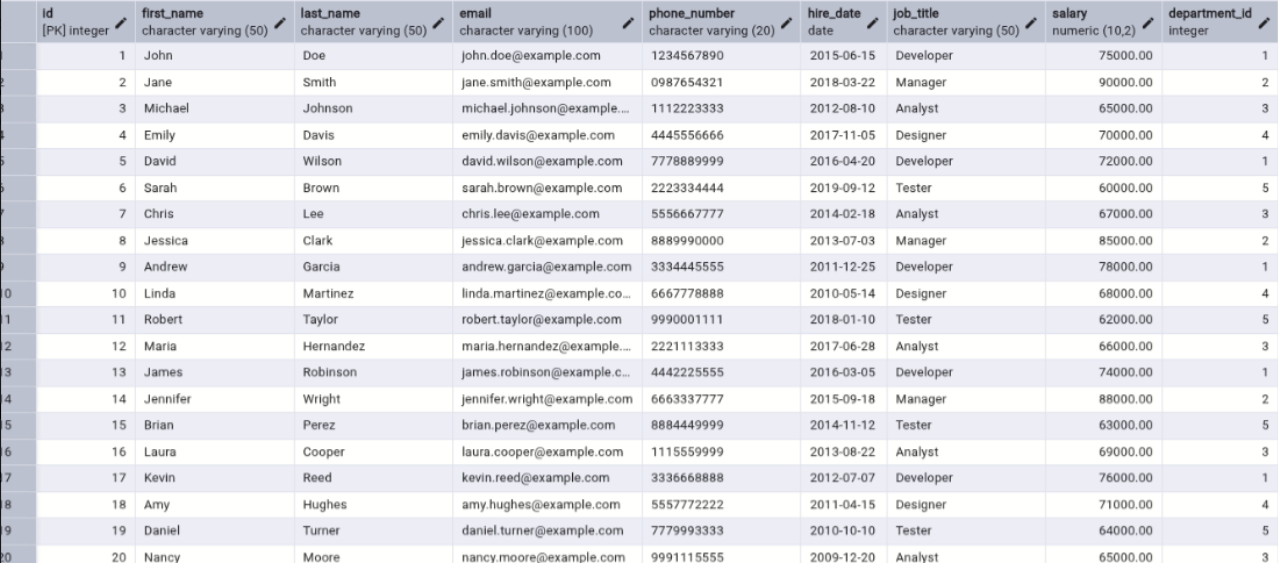


Рисунок 4 – Загрузка таблицы employees в PostgreSQL

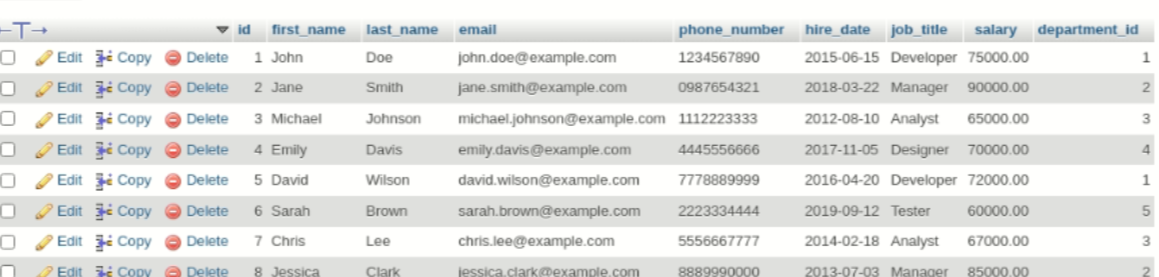


Рисунок 5 - Загрузка таблицы employees в MySQL

Далее была создана сама трансформация для перекачки сведений из PostgreSQL в MySQL, что продемонстрировано на рисунке 6.

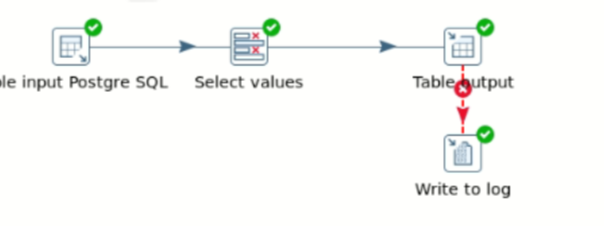


Рисунок 6 – Трансформация для переноса данных между базами

Job представлен на рисунке 7.

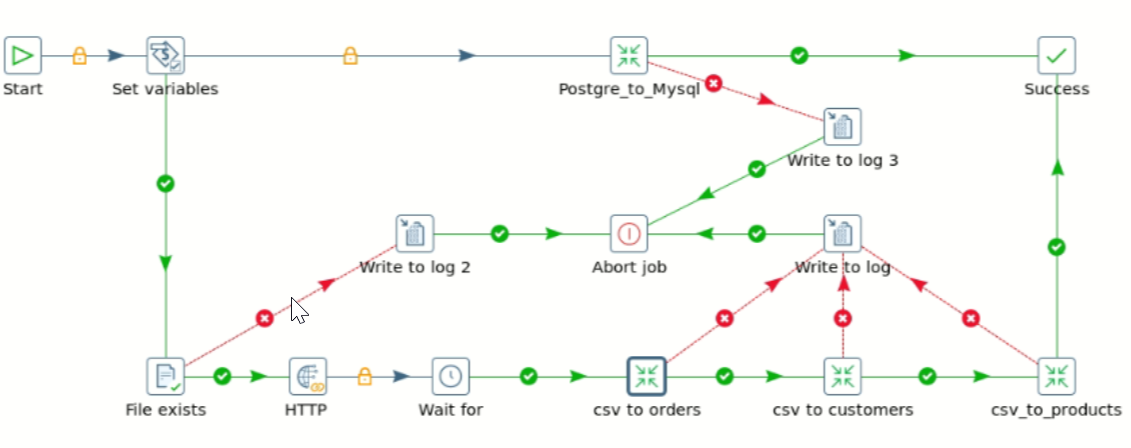


Рисунок 7 - Job

После данных шагов были созданы таблицы в рамках инд. задания на рисунках 8-13. Таблица со статусами содержит столбец, который фиксирует дату внесения изменений, как дополнение, которое не требовалось.

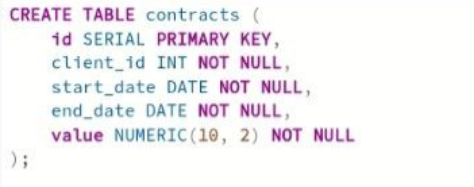


Рисунок 8 – Создание таблицы с контрактами в PostgreSQL

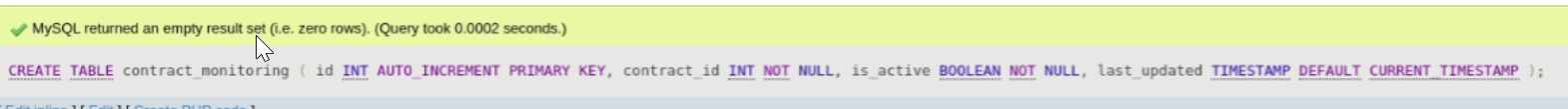


Рисунок 9 – Создание таблицы со статусами контрактов в MySQL



Рисунок 10 – Фрагмент таблицы с тестовыми данными по статусам

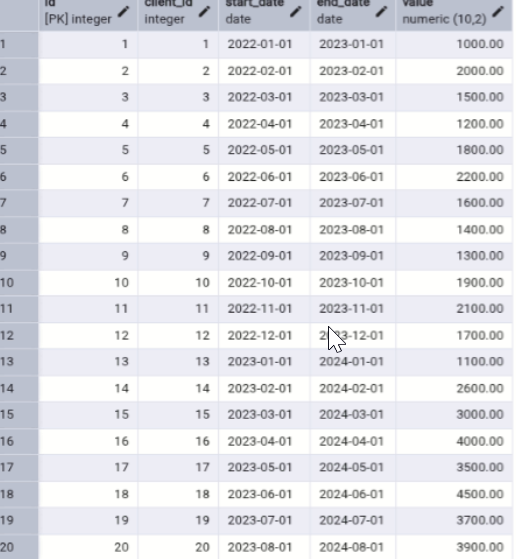


Рисунок 11 – Фрагмент таблицы с тестовыми данными по контрактам

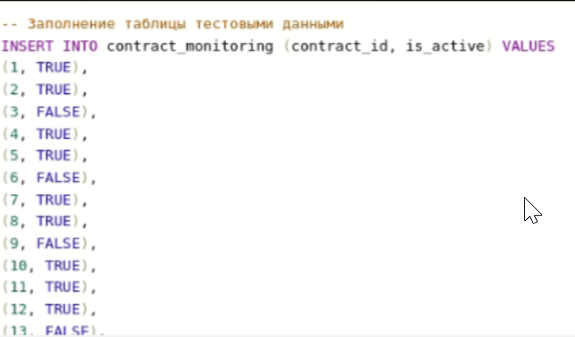


Рисунок 12 – Создание тестовых данных для статусов контрактов



Рисунок 13 – Создание тестовых данных для контрактов

После вышеперечисленных шагов была создана трансформация для обработки данных, в рамках нее для начала было выполнено подключение к PostgreSQL и MySQL, что можно увидеть на рисунках 14-15.

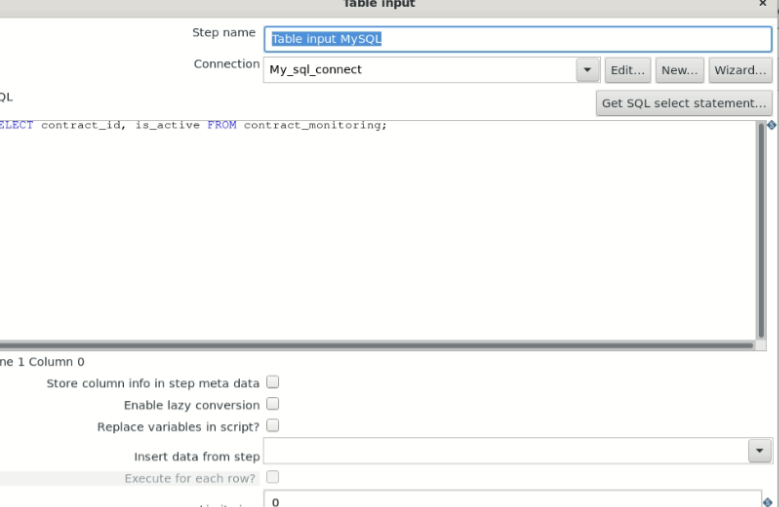


Рисунок 14 – Подключение к таблице со статусами контрактов

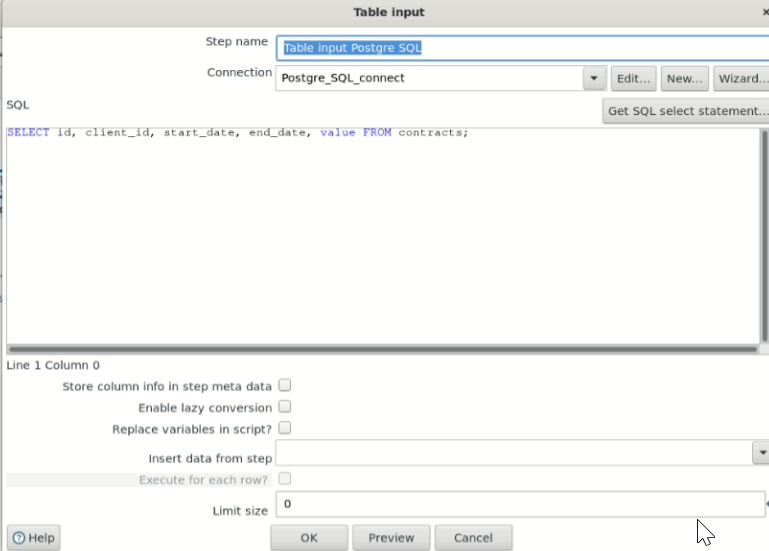


Рисунок 15 – Подключение к таблице с контрактами

Далее был добавлен Merge Join, объединяющий данные по столбцу на рисунке 16. Было выполнено внутреннее объединение по id контракта.

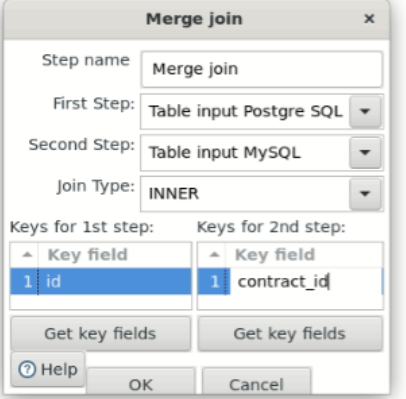


Рисунок 16 – Объединение 2 таблиц

Следующим шагом были отфильтрованы данные и выбраны только активные контракты на рисунке 17.

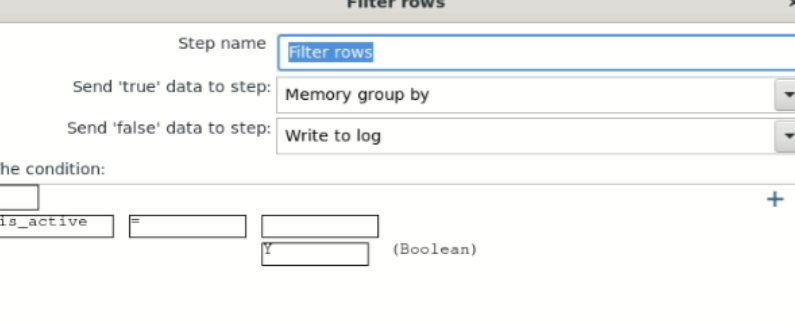


Рисунок 17 – Фильтрация для активных контрактов

Группировка по идентификатору контракта позволила посчитать общую сумму по каждому из контрактов на рисунке 18.

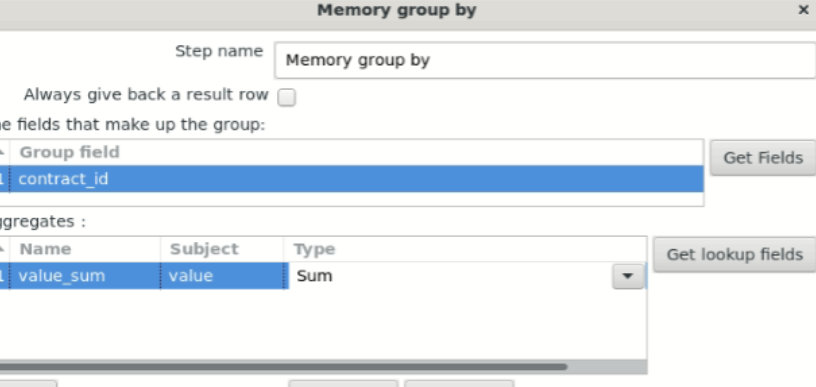


Рисунок 18 – Группировка по id контракта для расчета общей суммы

Объект калькулятора позволил рассчитать срок действия контракта на рисунке 19.

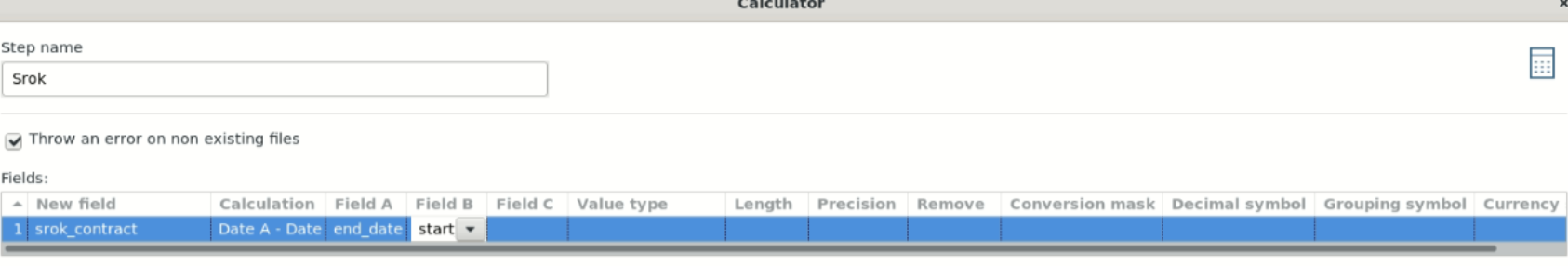


Рисунок 19 – Расчёт срока действия контракта

Для выгрузки результатов на рисунках 20-21 созданы таблицы для общей суммы по контрактам и для расчета срока действия контракта. Общая сумма контракта рассчитывалась для активных только контрактов.

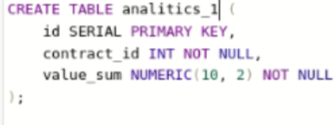


Рисунок 20 – Создание таблицы для первой аналитики

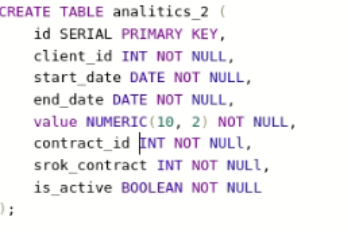


Рисунок 21 – Создание таблицы для второй аналитики

Был создан объект для загрузки аналитических отчетов в базу MySQL на рисунке 22 для первой аналитики, аналогичный для второй аналитики.

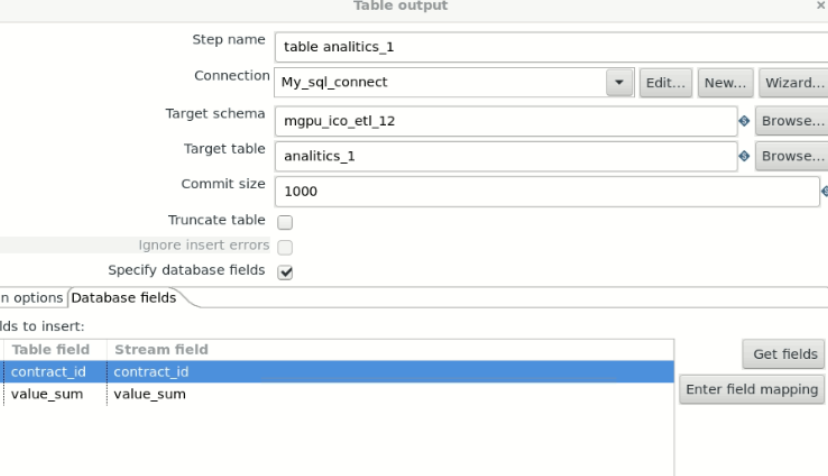


Рисунок 22 – Загрузка данных в базу для первой аналитики

С конечной схемой трансформации и с выгруженными данными можно ознакомиться на рисунках 23-25.

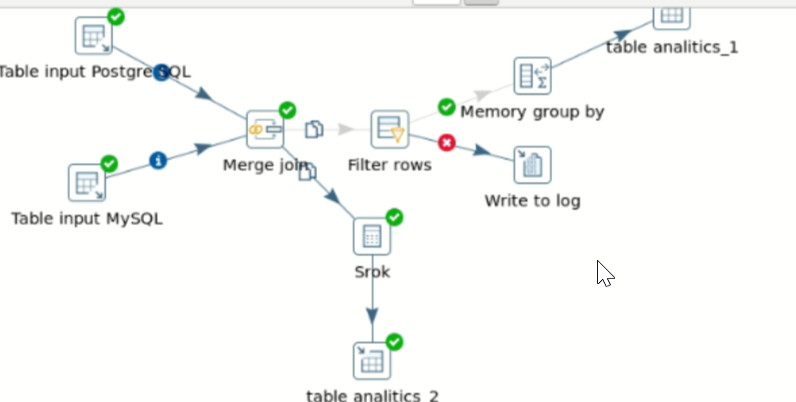


Рисунок 23 – Трансформация для контрактов

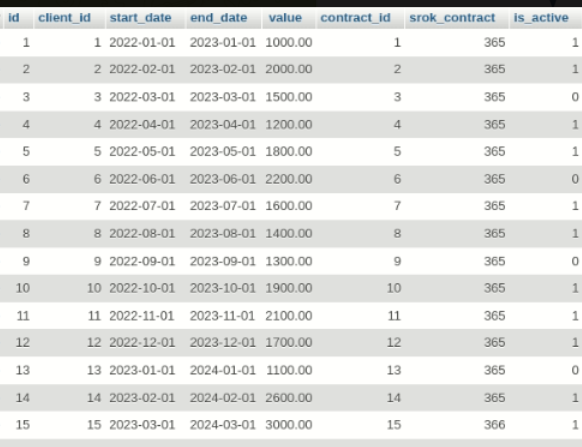


Рисунок 24 – Таблица по контрактам с рассчитанным сроком действия контракта

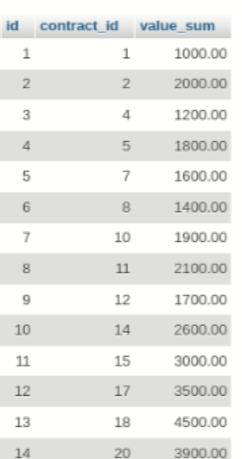


Рисунок 25 – Расчет общей суммы по активным контрактам

**Общий вывод:**

Цель и задачи были выполнены, а именно выполнены подключения и выгрузки в PostgreSQL и MySQL, изучены возможности слияния таблиц и работы с несколькими таблицами одновременно. Для выгрузки и расчетов PDI предлагает возможности работать с конкретными полями таблиц и выгрузкой конкретных полей. По результатам принцип работы с базами данных схож, можно с помощью SQL-запроса обратиться к базам и выбрать нужные данные, подключения работают схожим образом. Отличием от подключения к CSV файлам является то, что для файлов нужно указывать путь, а не настраивать подключение. В рамках самостоятельного задания был реализован пример, созданы таблицы по контрактам и их статусам, была выполнена фильтрация и отобраны активные заказы, для них был выполнен расчет суммы по id контракта, по всем контрактам был рассчитан срок действия, результаты выгружены в базу MySQL.