|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Базовая кафедра №234 — Управляющих ЭВМ

**Отчет по ознакомительной практике**

|  |  |
| --- | --- |
| **Выполнил:**  Студент группыИКМО-05-23 | Миронов Д.С. |
| **Проверил:** | Прилипко В.А. |

Москва, 2024

Оглавление

[Введение 3](#_Toc166423358)

[Выполнение практического задания 3](#_Toc166423359)

[Функциональное решение 4](#_Toc166423360)

[Как формируется куб 5](#_Toc166423361)

[Как формируются индексы 6](#_Toc166423362)

[Получение разворота OLAP-куба по индексам 7](#_Toc166423363)

[Заключение 8](#_Toc166423364)

[Литература 8](#_Toc166423365)

# Введение

Страховая компания «Согаз» — крупная российская страховая компания, отнесена к разряду системообразующих, лидер одноимённой страховой группы Согаз.

В качестве Python-программиста в данной компании мне назначаются задачи связанные с разработкой автоматизации внутрикорпоративных систем.

# Выполнение практического задания

Одной из задач, стало построение системы позволяющую быстро внедрять аналитические методы и принимать решения на основе данных.

При создании системы для OLAP – кубов[1] нужно учитывать две основные проблемы, это потребляемая память при расчете каждой агрегации куба и скорость расчета всех агрегаций. Поскольку при каждом последующем расчете агрегаций, объем куба увеличивает в несколько раз.

На текущий момент есть два основных решения в области BI – Hyperion planning и Qlik sense. Оба решения предлагают ограниченный функционал и высокую стоимость владения. Детально прописывая план внедрения платформы, многие компании сходятся во мнении [2], что быстрее и дешевле создать свою платформу, используя более современные инструменты анализа данных, разработанные для DS (data science) и открытые библиотеки для визуализации для современных фреймворков JavaScript.

В данной статье описывается построение системы позволяющую быстро внедрять аналитические методы и принимать решения на основе данных. В виду того что в создаваемой системе основной функционал будет построен на создании OLAP – куба.

# Функциональное решение

Перед началом формирование OLAP - куба, необходимо создать его структуру (рис. 1), то из чего он будет состоять. Основой, конечно же являются данные и стороны. Стороны — это измерение куба, то что будет группироваться, при формировании куба.

Необходимо указать столбцы основного файла и их иерархию. Пример иерархии или же одной стороны – «месяц – неделя – год». Данная сторона будет называться в структуре, например, «Дата». Таких сторон в кубе может быть не ограниченное количество, но с каждой добавляемой стороной и глубины иерархии, увеличивается объем куба и сложность при его расчете. В качестве данных необходимо указать столбцы с числовыми значениями на основе которых буду проводиться расчеты.

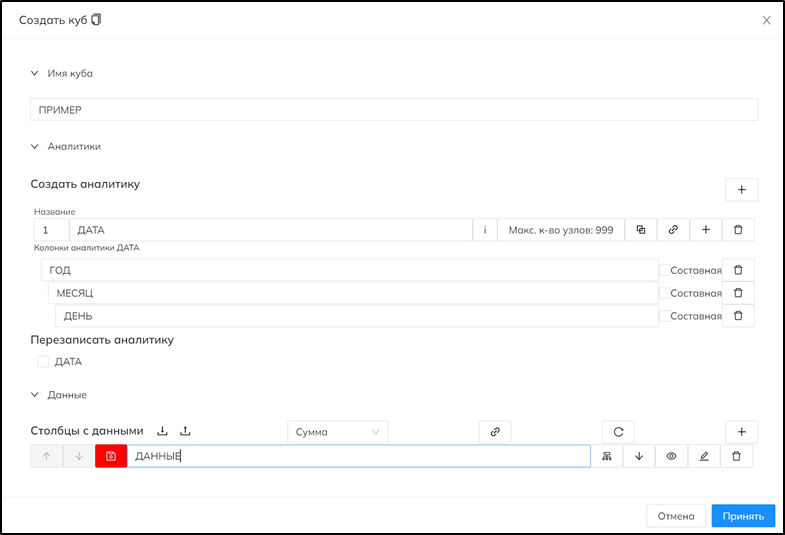


Рис. 1. Формирование структуры [разработано автором]

После описания сторон и данных куба, формируются параметры каждой стороны и столбов с данными, для того чтобы эффективно хранить полученные значения и быстро выводить данные при запросе.

Для каждой стороны в структуре должна находится следующая информация: название стороны, названия столбцов иерархии стороны, порядковый номер стороны, длина индекса для данной стороны.

# Как формируется куб

Основной идеей формирование куба, заключается в том, чтобы формировать каждый последующий разворот куба мы будет на основе предыдущего, так идя он начальной таблице, мы постепенно будет ее увеличивать, добавляя сгруппированные значения и из полученной таблице будем рекурсивно группировать её по каждой иерархии стороны (рис. 2).

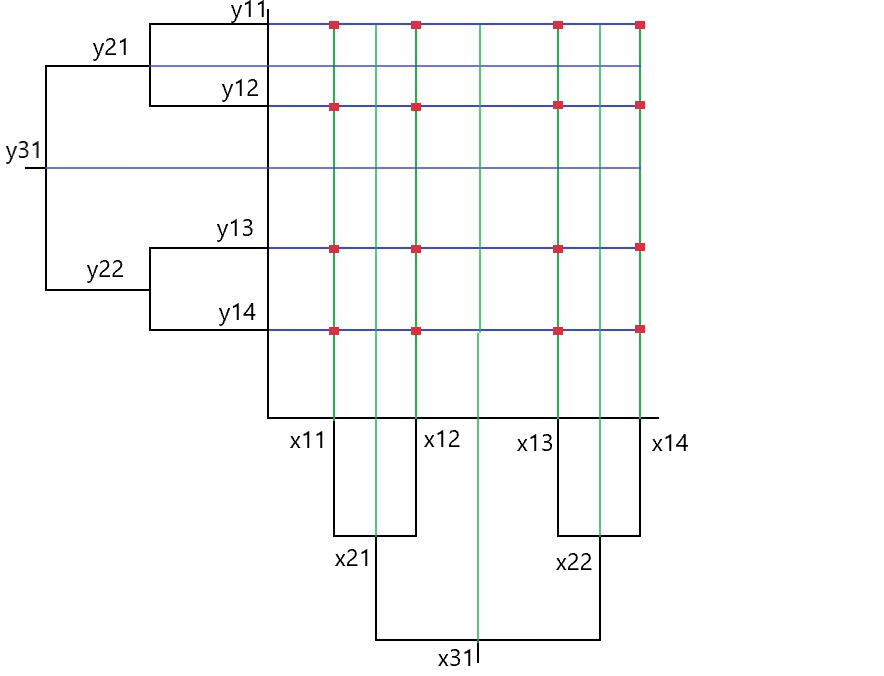


Рис. 2. Агрегация в двумерной плоскости [разработано автором]

На рис. 2 представлена первая агрегация, при которой формируются значения на листьях. Добавляя полученные значения к основным, мы можем делать группировку следующего уровня иерархии стороны. Нам не приходится каждый раз рассчитывать куб до нужного уровня, поскольку при каждой агрегации уровня иерархии стороны, сформированные данные уже будут, но используя больше памяти при каждой последующей агрегации над кубом. Данную проблему можно решить тем, что каждый столбец данных считать отдельно, последовательно. Таким образом если у нас в изначальных столбцах данных находятся 100 столбцов, мы соберем куб 100 раз для каждого столбца, следовательно, мы уменьшим объем потребляемой памяти при формировании куба, но скорость формировании всего куба увеличиться.

# Как формируются индексы

С точки зрения математики для того чтобы хранить плоскость на прямой необходимо каждой точки дать свой уникальный индекс (номер), таким образов если у нас многомерная плоскость, то для каждой из плоскостей нужно присвоить свой индекс. Сформированный куб это и есть многомерная пространство плоскостей. Каждый индекс будет состоять из индексов стороны.

Для того чтобы сформировать индекс стороны, нужно подготовить таблицу, в которой они будут храниться. Данная таблица будет состоять из уникальных значений каждого уровня иерархии стороны, в которой в столбце «Имя» будут находится имена всех уникальных значений стороны, а в столбце «Индекс» число для каждого уникального значения в столбце «Имя».

Перед подстановкой индексов, нужно отредактировать с агрегированные данные. Для этого нужно создать столбы с названиями всех сторон и подставить первое не пустое значение из названий столбцов иерархии сторон. Таки образов в итоговой таблицу окажется количество столбцов равное количеству сторон.

Далее необходимо заменить значения в столбцах на значение в таблице с индексами и соединить каждую строку в одну ячейку. Таким образом получаться индексы для каждого значения OLAP-куба.

# Получение разворота OLAP-куба по индексам

Для того чтобы получить запрашиваемый разворот, необходимо сделать запрос, в котором указан индекс запрашиваемой стороны, если запрашиваемый разворот включает в себя две стороны, то в запросе нужно указать два индекса.

Приведем пример (рис. 3), для каждой стороны присвоен индекс, а максимальная длина каждого два знака, количество символов зависит от количества уникальных значений в столбце стороны, «ID чел» - 1, «ДАТА» - 2, «Данные» - 0 и 1, столбец с данными в индексе всегда стоит последним и должен иметь минимум два значения индекса, для суммы всех столбцов с данными (это индекс 0) и для каждого столбца индекс становится плюс 1. Так как в примере всего лишь один столбец с данными, индекс будет выглядеть следующем образом , на рисунке 3, представлен разворот с индексами 000100 и 010100.

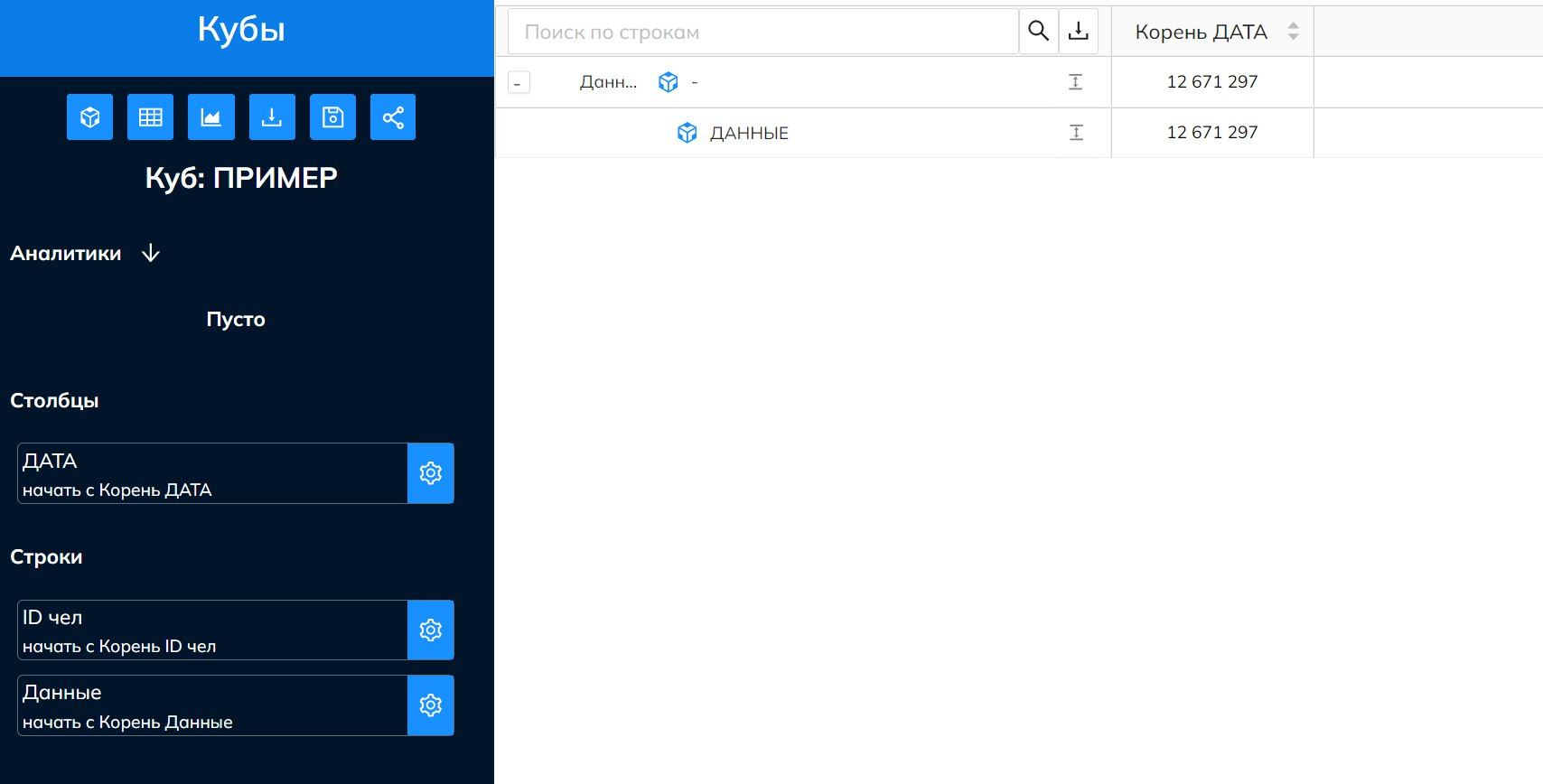


Рис. 3. Разворот куба [разработано автором]

Если продолжить получать развороты, то мы увидим все больше данных (рис. 4).

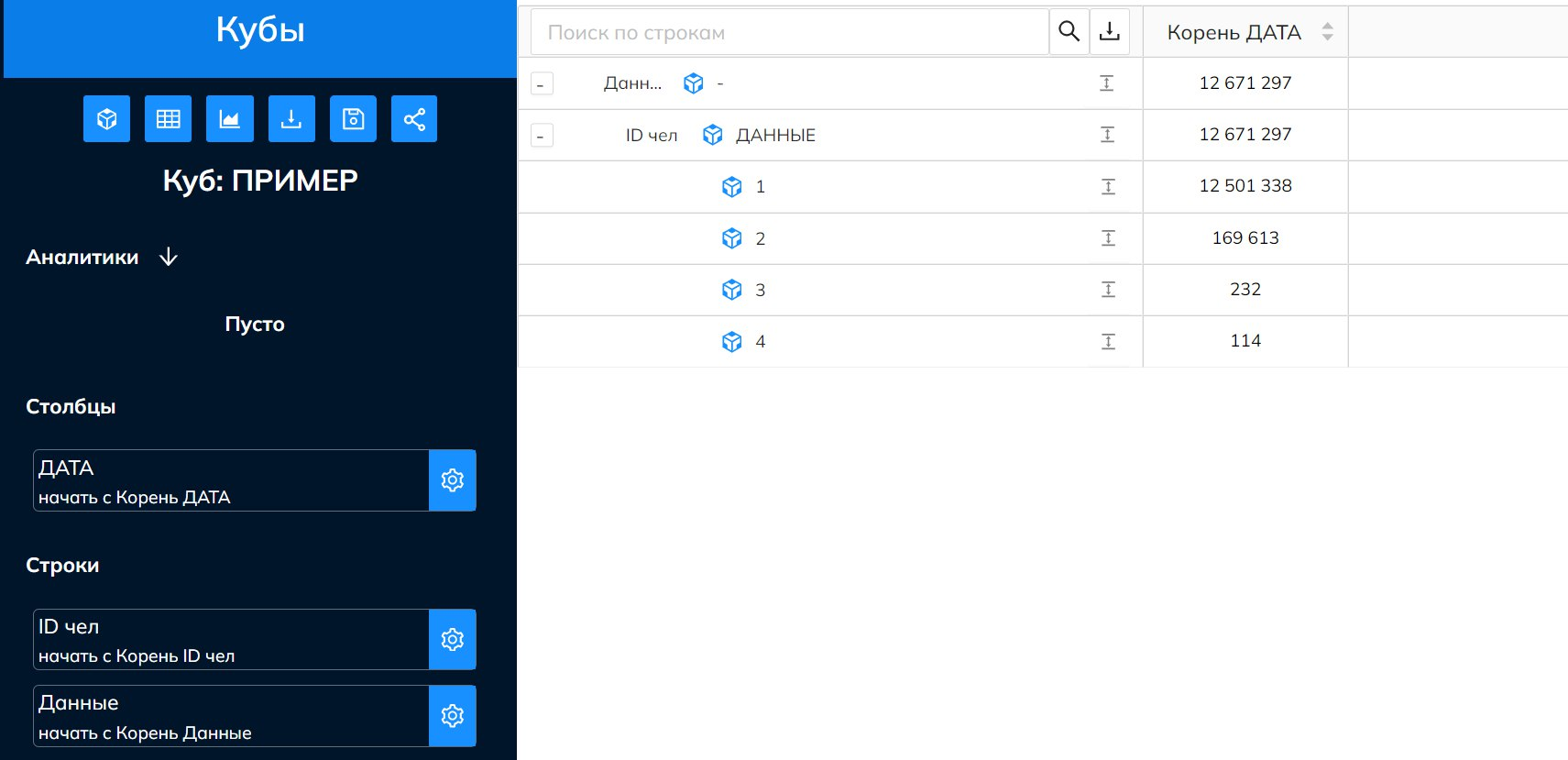


Рис. 4. Разворот куба [разработано автором]

Так же может понадобиться рассмотреть столбец «ДАТА» более детально, то нажав на соответственный столбец, появятся уровни данной стороны (рис. 5).

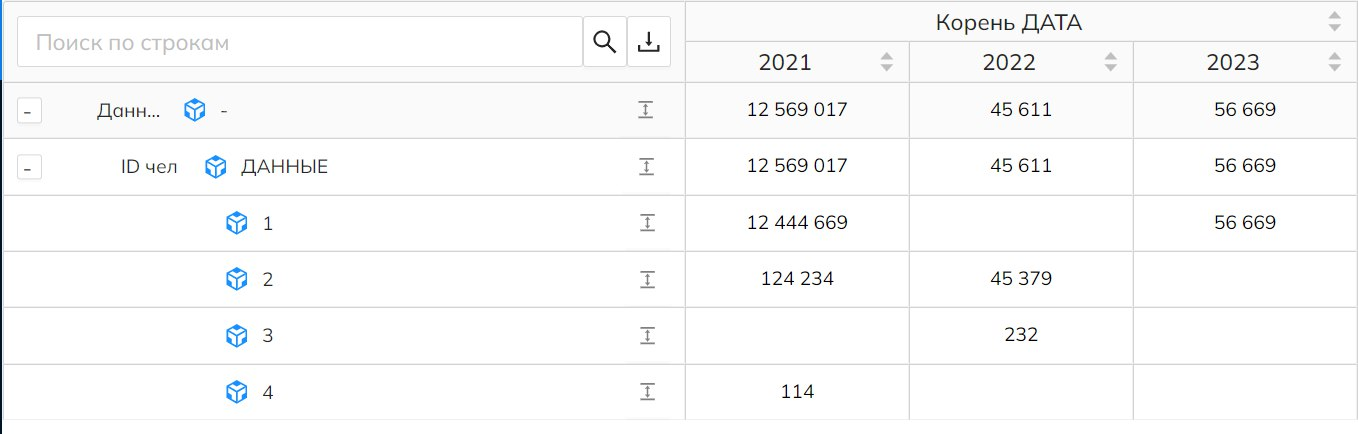


Рис. 5. Разворот куба с детальным разворотом столба [разработано автором]

# Заключение

По итогам выполнения работы была построена система, позволяющая быстро внедрять аналитические методы и принимать решения на основе данных. Данная система успешно внедрена внутрь компании.

Были так же изучены язык программирования JavaScript и фреймворк React.

# Литература

1. Введение в OLAP и многомерные базы данных — URL: http://www.olap.ru/basic/alpero2i.asp (дата обращения: 25.11.2023).
2. «I Was Seduced By a Build Scenario»: 11 Ways to Avoid This Exec’s Greatest Tech Failure//Better Cloud. — URL: https://www.bettercloud.com/monitor/build-vs-buy/ (дата обращения: 23.11.2023).