УНИВЕРСИТЕТ ИТМО

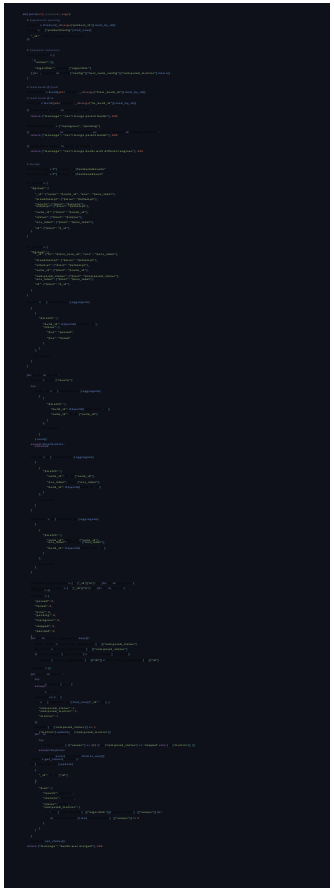
# Автоматизация индексирования базы данных на основе истории запросов

Кульбако Артемий Юрьевич, P34112

Гаврилов Антон Валерьевич, преподаватель ПИиКТ

# Проблема

Старые базы данных могут содержать очень сложные и непонятные запросы, скорость выполнения которых необходимо увеличить. Ручная оптимизации займёт много времени и напрямую зависит от опыта работы сотрудника с данной даталогической моделью.



Пример запроса к БД из рабочего проекта на 200+ строк, который тут даже не разглядеть.

# Задачи и цели

**Цели:**

- Минимизировать время на оптимизацию БД пользователем
- Максимально увеличить скорость выполнения запросов к БД

**Задачи:**

- Обзор существующих решений
- Разработка требований, предъявляемых к создаваемому приложению
- Выбор средств реализации
- Проектирование и создание тестовой базы данных
- СОЗДАНИЕ ПРИЛОЖЕНИЯ АВТОИНДЕКСИРОВАНИЯ БАЗЫ ДАННЫХ
- Тестирование приложения
- Анализ полученных результатов

# Обзор аналогов

8 из 10 самых популярных* СУБД подобной функциональности не имеют.

| Рейтинг | | | СУБД | Модель данных | Счёт | | |
|---|---|---|---|---|---|---|---|
| Май 2022 | Апрель 2022 | Май 2021 | | | Май 2022 | Апрель 2022 | Май 2021 |
| 1. | 1. | 1. | Oracle | Реляционная, Мультимодельная | 1262.82 | +8.00 | -7.12 |
| 2. | 2. | 2. | MySQL | Реляционная, Мультимодельная | 1202.10 | -2.06 | -34.28 |
| 3. | 3. | 3. | Microsoft SQL Server | Реляционная, Мультимодельная | 941.20 | +2.74 | -51.46 |
| 4. | 4. | 4. | PostgreSQL | Реляционная, Мультимодельная | 615.29 | +0.83 | +56.04 |
| 5. | 5. | 5. | MongoDB | Документная, Мультимодельная | 478.24 | -5.14 | -2.78 |
| 6. | 6. | ↑ 7. | Redis | Ключ-значение, Мультимодельная | 179.02 | +1.41 | +16.85 |
| 7. | ↑ 8. | ↓ 6. | IBM Db2 | Реляционная, Мультимодельная | 160.32 | -0.13 | -6.34 |
| 8. | ↓ 7. | 8. | Elasticsearch | Поисковой движок, Мультимодельная | 157.69 | -3.14 | +2.34 |
| 9. | 9. | ↑ 10. | Microsoft Access | Реляционная | 143.44 | +0.66 | +28.04 |
| 10. | 10. | ↓ 9. | SQLite | Реляционная | 134.73 | +1.94 | +8.04 |



* по версии db-engines.com

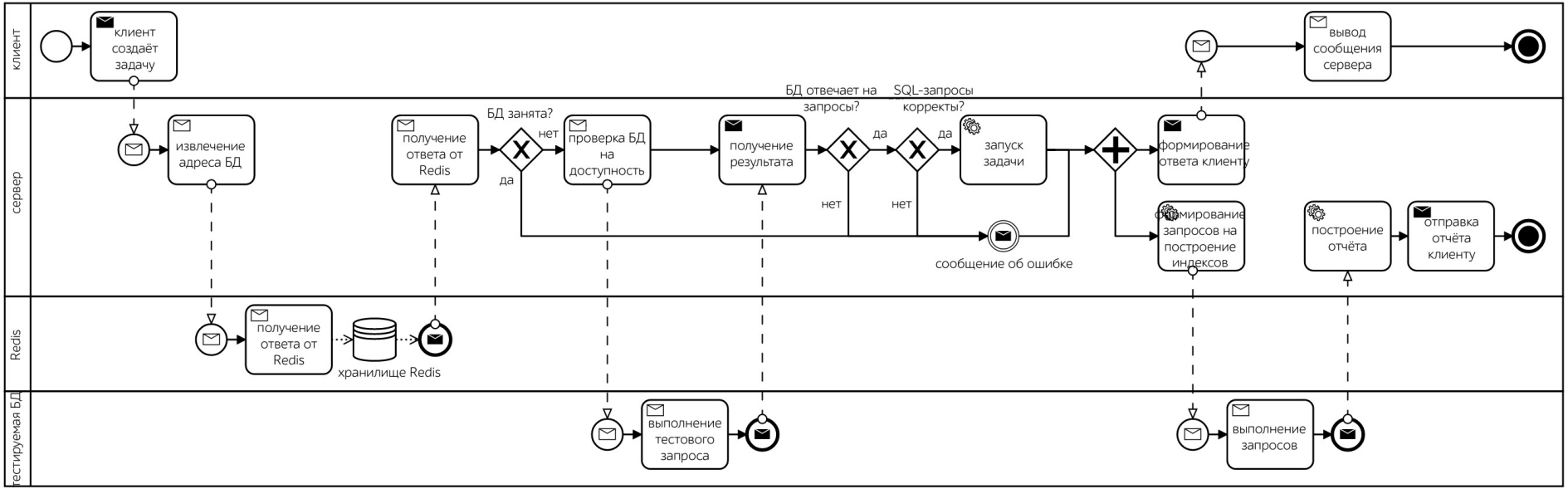
Oracle Database и Microsoft SQLServer имеют функцию автоиндексирования, но это дорогие коммерческие продукты, а сам процесс представляет собой «черный ящик».

Нельзя официально приобрести в России.

**Преимущества моего решения:**

- Поддержка любой БД с JDBC-драйвером
- Контролируемые процесс (формирование отчётов)
- Открытый исходный код.

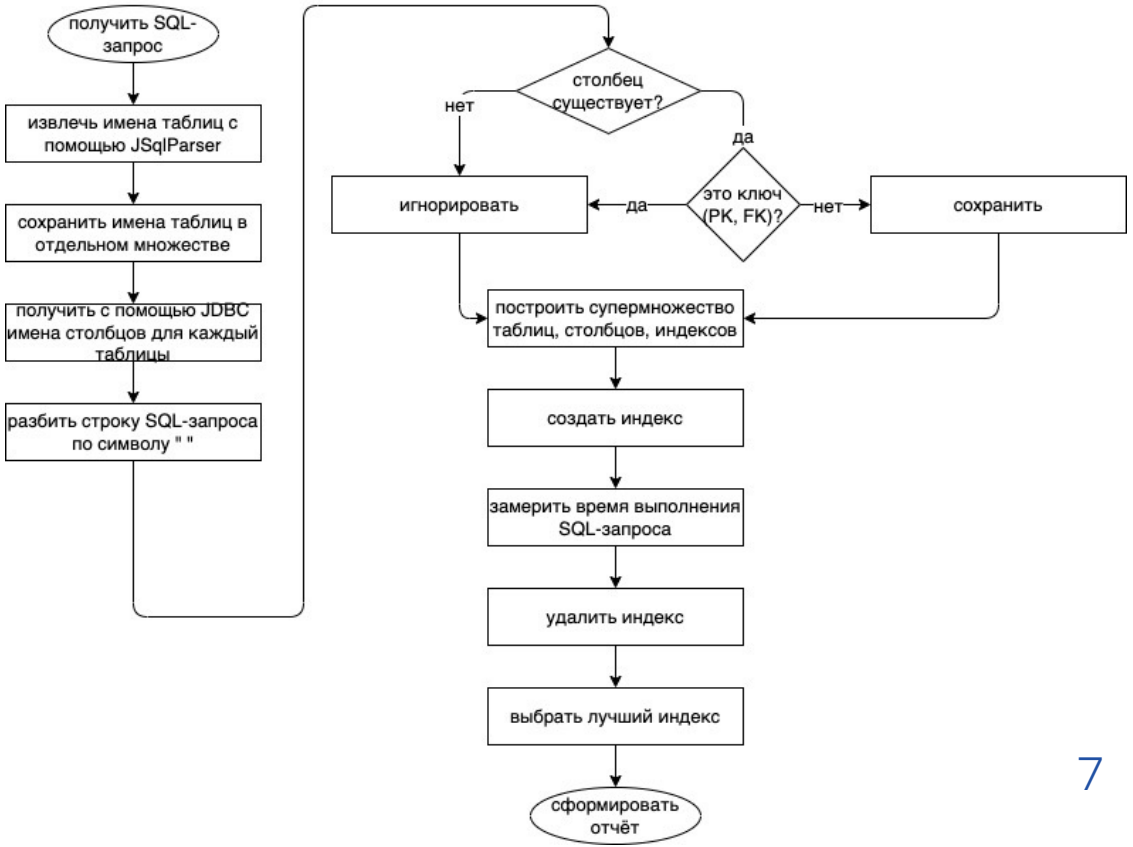
# Рабочий конвейер



# Алгоритм формирования индексов

Программа создаёт для каждого уникального SQL-запроса из истории все возможные комбинации:

«Таблица-поле-тип индекса»

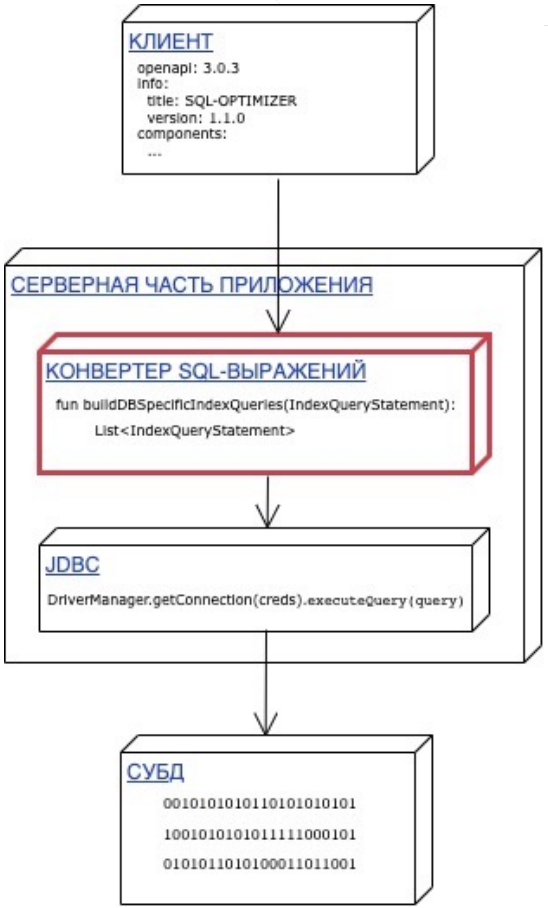


# Интеграция СУБД

За трансформацию выражений из ANSI SQL в специфичный для СУБД SQL отвечает модуль конвертации.

Для добавления поддержки новой СУБД необходимо переопределить ОДИН метод:

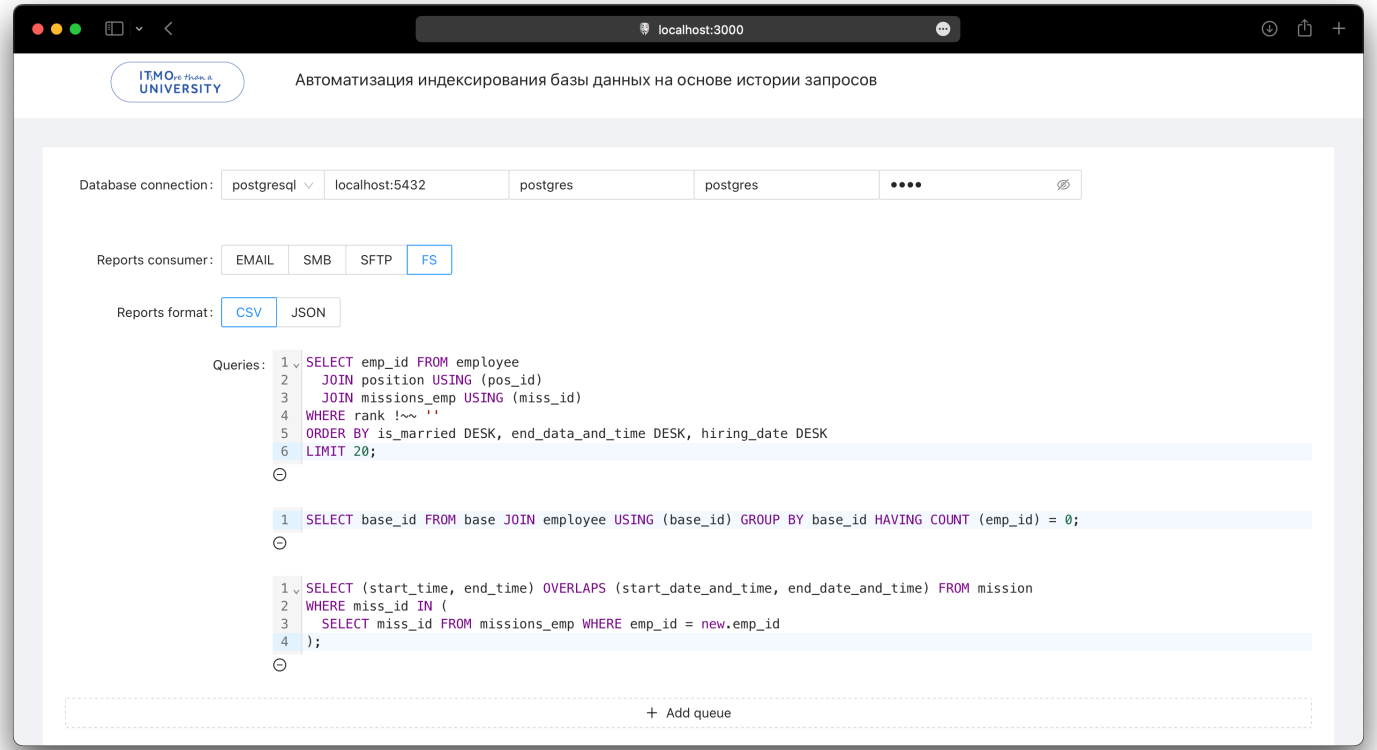
```
private fun formIndexesQueries():
    List<IndexQueryStatement>
```



УНИВЕРСИТЕТ ИТМО

# Графический интерфейс

Приложение для автоматизации индексирования БД




ITMO re than a UNIVERSITY

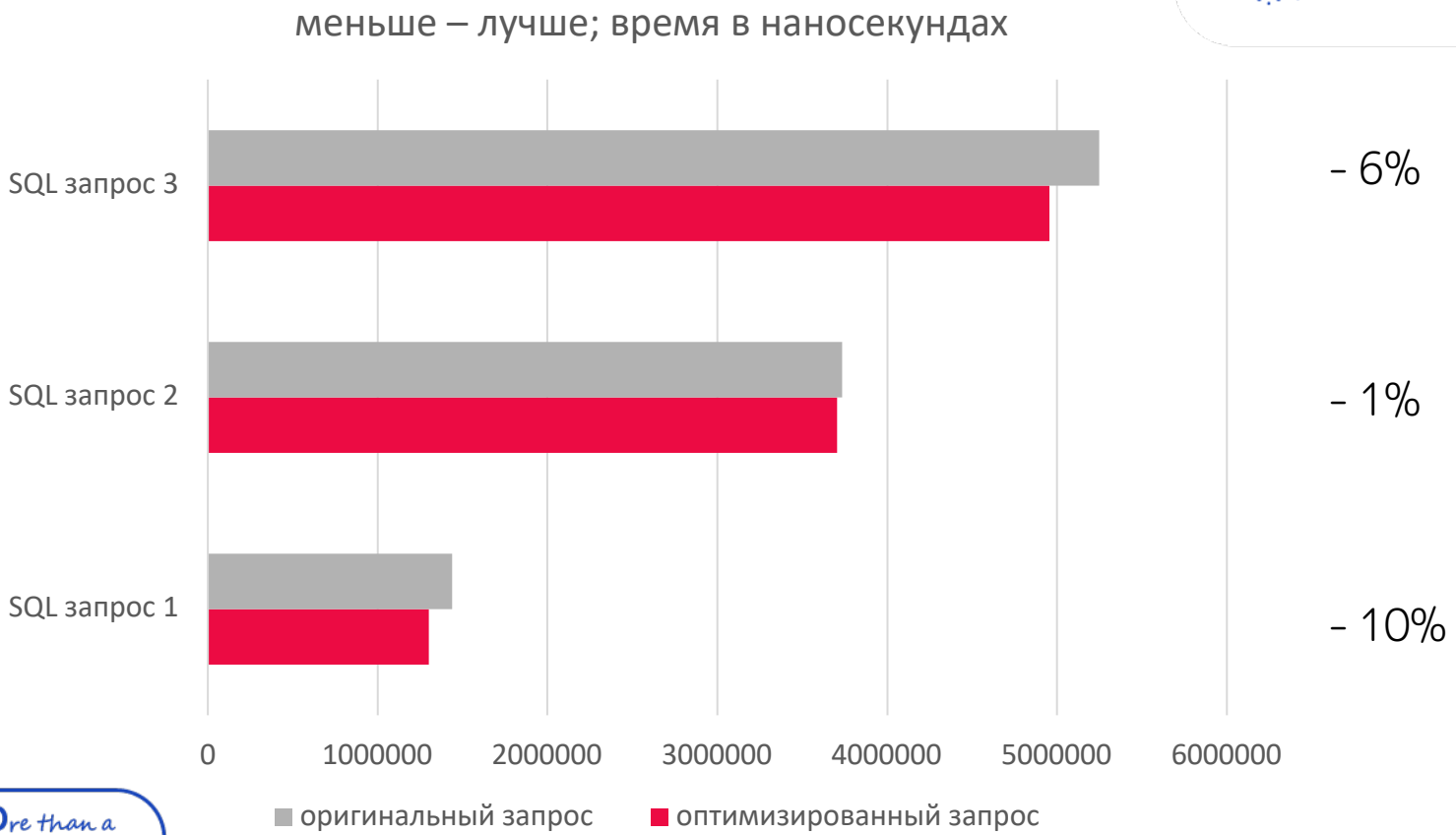
# Результаты

Запрос:

```
SELECT emp_id FROM employee JOIN position USING (pos_id) JOIN
missions_emp USING (emp_id) JOIN mission USING (miss_id) WHERE rank !~~
'' ORDER BY is_married DESC, end_date_and_time DESC, hiring_date
DESCLIMIT 20;
```

Отчёт:

| | indexStatement | diff | timeTaken |
|---|---|---|---|
| 1 | CREATE INDEX Is_marriedEmployeeHASH ON employee USING HASH(is_married); | 173166 | 4956042 |
| 2 | CREATE INDEX Is_marriedHiring_dateEmployeeBTREE ON employee USING BTREE(is_married, hiring_date); | 123791 | 5005417 |
| 3 | CREATE INDEX End_date_and_timeMissionBTREE ON mission USING BTREE(end_date_and_time); | 78250 | 5050958 |
| 4 | CREATE INDEX Emp_idIs_marriedHiring_dateEmployeeBTREE ON employee USING BTREE(emp_id, is_married, hiring_date); | 71375 | 5057833 |
| 5 | CREATE INDEX RankPositionHASH ON position USING HASH(rank); | 45833 | 5083375 |


меньше – лучше; время в наносекундах
SQL запрос 3
SQL запрос 2
SQL запрос 1
- 6%
- 1%
- 10%
0
1000000
2000000
3000000
4000000
5000000
6000000
оригинальный запрос
оптимизированный запрос

# Выводы

- Приложение выполняет поставленные задачи
  - Минимизирует время, затрачиваемое на оптимизацию базы пользователем (программа выполняет индексацию автоматически)
  - Ускоряет запросы к базе данных
- Расширяемая архитектура позволяет легко добавлять новые СУБД
- Мои навыки и компетенции были улучшены

# Спасибо за внимание! Готов к вашим вопросам.

www.ifmo.ru

ITMO more than a UNIVERSITY