Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ_____

КАФЕДРА _____СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ_____

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

Анализ времени выполнения типовых запросов в PostgreSql и ClickHouse

Студент __ИУ5-33М_____ _____ **С.С. Алехин**
(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель _____ **Ю.Е. Гапанюк**
(Подпись, дата) (И.О.Фамилия)

2023 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ
Заведующий кафедрой ___ИУ5____
(Индекс)
_____ ___В.И. Терехов__
(И.О.Фамилия)
«_04_» ____сентября_____ 2023 г.

З А Д А Н И Е
на выполнение научно-исследовательской работы

по теме ___Анализ времени выполнения типовых запросов в PostgreSQL и ClickHouse_____
_____

Студент группы _ИУ5-33М__

_____Алехин Сергей Сергеевич_____
(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)
_____ИССЛЕДОВАТЕЛЬСКАЯ_____
Источник тематики (кафедра, предприятие, НИР) _____КАФЕДРА_____

График выполнения НИР: 25% к ____ нед., 50% к ____ нед., 75% к ___ нед., 100% к ____ нед.

Техническое задание __Разработать и сравнить типовые запросы в PostgreSQL и ClickHouse_____
_____
_____
_____
_____

Оформление научно-исследовательской работы:

Расчетно-пояснительная записка на __29__ листах формата А4.
Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)
_____
_____
_____
_____

Дата выдачи задания «_04_» ____сентября_____ 2023 г.

Руководитель НИР _____ _____Ю.Е. Гапанюк__
(Подпись, дата) (И.О.Фамилия)

Студент _____ **С.С. Алехин**
(Подпись, дата) (И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

Оглавление

Введение

В современном информационном обществе огромное значение приобретают системы управления базами данных, обеспечивающие эффективное хранение, обработку и извлечение данных. PostgreSQL и ClickHouse представляют собой два ведущих реляционных базы данных, успешно применяемых в различных областях, от веб-разработки до аналитики больших объемов данных.

Одним из критически важных аспектов производительности баз данных является время выполнения запросов. Данная курсовая работа посвящена анализу времени выполнения типовых запросов в PostgreSQL и ClickHouse с целью выявления особенностей их работы, эффективности и применимости в различных сценариях использования. Путем изучения и сравнения характеристик выполнения запросов в обеих системах мы стремимся выявить преимущества каждой из них в зависимости от конкретных задач и контекста использования.

Основная часть

Структура данных

Для анализа возьмем список вакансий из HeadHunter. На основе этих данных будем проводить анализ и сравнение. Пример вакансии


```
{
 "id": "75000001",
 "premium": false,
 "billing_type": {
  "id": "standard",
  "name": "Стандарт"
 },
 "relations": [],
 "name": "Велокурьер в компанию — партнёр Яндекса (м.Елизаровская)",
 "insider_interview": null,
 "response_letter_required": false,
 "area": {
  "id": "2",
  "name": "Санкт-Петербург",
  "url": "https://api.hh.ru/areas/2"
 },
 "salary": {
  "from": null,
  "to": 75000,
  "currency": "RUR",
  "gross": true
 },
 "type": {
  "id": "open",
  "name": "Открытая"
 },
 "address": {
  "city": "Санкт-Петербург",
  "street": "улица Ткачей",
  "building": "6",
  "lat": 59.893809,
  "lng": 30.432757,
  "description": null,
  "raw": "Санкт-Петербург, улица Ткачей, 6",
  "metro": null,
  "metro_stations": []
 },
```

```
 "allow_messages": true,
 "experience": {
  "id": "noExperience",
  "name": "Нет опыта"
 },
 "schedule": {
  "id": "fullDay",
  "name": "Полный день"
 },
 "employment": {
  "id": "full",
  "name": "Полная занятость"
 },
 "department": null,
 "contacts": null,
 "description": "<p>Партнер сервиса <strong>Яндекс Лавка</strong> приглашает к сотрудничеству Велокурьеров .</p> <p><strong>Что нужно делать:</strong></p> <ul> <li>Забирать заказы и вовремя доставлять их клиентам.</li> </ul> <p><strong>Условия:</strong></p> <p>— Ты сам выбираешь где и когда доставлять заказы - утром, днем, или в удобное для тебя время. Удобный график ты выбираешь сам в комфортном приложении;</p> <ul> <li>Своевременные еженедельные выплаты дохода;</li> <li><strong>Почасовой доход вне зависимости от числа заказов (повышенные бонусы за час);</strong></li> <li><strong>Бонусы: за каждый выполненный заказ + чаевые от клиентов</strong></li> </ul> <p>— Велосипед и сумка предоставляются <strong>бесплатно</strong> .<br /><br />— <strong>Бонус за друзей! Зовите друзей 1 друг = до 10 000 рублей, 5 друзей до 50 000 рублей!</strong></p> <p>— Комнаты отдыха: Где вы можете зарядить телефоны, переодеться, выпить чай и покушать;</p> <p>— Возможность выбрать район доставки;</p> <p>— Возможность выбрать день и время доставок;</p> <p>— Доставка на небольшие расстояния;<br /><br /><strong>Требования:</strong></p> <p>— Доброжелательность и пунктуальность;</p> <p>— Смартфон на базе Android;</p> <p>— Способность ориентироваться по картам в телефоне;</p> <p>— Знание основных правил дорожного движения.</p> <p>Наша вакансия также подойдет вам, если вы искали: с еженедельной оплатой,доставка выходного дня, свободное время выхода на слот, гибкое расписание,доставка, доставка документов посылок, доставка в районе, доставка с домом, без опыта, начинающий специалист, выходные, в вечернее время, яндекс доставка, самокат, яндекс еда, яндекс лавка, яндекс про, яндекс маркет, сбер маркет,озон, магнит, пятёрочка, перекресток, дикси, лента, ашан, глобус, окей, детский мир, спорт мастер, летуаль, леруа мерлен, деливери клаб, delivery club, dostavista, достависта, dodo, додо, сдек</p> <p> </p>",
 "branded_description": null,
 "vacancy_constructor_template": null,
```

```
"key_skills": [
  {
    "name": "Пользователь ПК"
  },
  {
    "name": "Грамотная речь"
  },
  {
    "name": "Работа в команде"
  },
  {
    "name": "Телефонные переговоры"
  },
  {
    "name": "Деловое общение"
  }
],
"accept_handicapped": false,
"accept_kids": false,
"archived": true,
"response_url": null,
"specializations": [],
"professional_roles": [
  {
    "id": "58",
    "name": "Курьер"
  }
],
"code": null,
"hidden": false,
"quick_responses_allowed": false,
"driver_license_types": [],
"accept_incomplete_resumes": false,
"employer": {
  "id": "5974128",
  "name": "Партнер сервиса Яндекс.Лавка",
  "url": "https://api.hh.ru/employers/5974128",
  "alternate_url": "https://hh.ru/employer/5974128",
  "logo_urls": {
    "90": "https://hhcdn.ru/employer-logo/4173508.png",
    "240": "https://hhcdn.ru/employer-logo/4173509.png",
    "original": "https://hhcdn.ru/employer-logo-original/933239.png"
  },
  "vacancies_url": "https://api.hh.ru/vacancies?employer_id=5974128",
  "trusted": true
```

```
  },
  "published_at": "2023-01-04T23:53:54+0300",
  "created_at": "2023-01-04T23:53:54+0300",
  "initial_created_at": "2023-01-04T23:53:54+0300",
  "negotiations_url": null,
  "suitable_resumes_url": null,
  "apply_alternate_url":
"https://hh.ru/applicant/vacancy_response?vacancyId=75000001",
  "has_test": false,
  "test": null,
  "alternate_url": "https://hh.ru/vacancy/75000001",
  "working_days": [
    {
     "id": "only_saturday_and_sunday",
     "name": "Работа только по сб и вс"
    }
  ],
  "working_time_intervals": [
    {
     "id": "from_four_to_six_hours_in_a_day",
     "name": "Можно работать сменами по 4–6 часов в день"
    }
  ],
  "working_time_modes": [
    {
     "id": "start_after_sixteen",
     "name": "Можно начинать работать после 16:00"
    }
  ],
  "accept_temporary": true,
  "languages": []
}
```


Для проектирования схемы базы данных вначале рассмотрим пример данных, которые будут приходить из HH. Выделим основные атрибуты, которые будут нам полезны:

- `id` - идентификатор вакансии
- `name` - название вакансии
- `premium` - флаг премиум вакансии
- `salary_from` - зарплата от
- `salary_to` - зарплата до

- `salary_currency` - зарплата валюта
- `salary_gross` - флаг до вычета(1)/чистыми(0)
- `vacancy_type` - тип вакансии (открытая, закрытая, архивная и т.д.)
- `experience` - требуемый опыт (нет опыта, от 1 до 2 лет и т.д.)
- `schedule` - график работы (Удаленная работа, Гибкий график и т.д.)
- `employment` - занятость (Частичная занятость, Полная занятость и т.д.)
- `description` - описание вакансии
- `key_skills` - ключевые навыки
- `professional_roles` - роль профессии
- `employer_id` - id нанимателя
- `employer_name` - название нанимателя
- `employer_url` - ссылка на профиль нанимателя
- `employer_logo_url` - ссылка на аватар нанимателя
- `employer_vacancies_url` - ссылка на вакансии нанимателя
- `trusted_employer` - флаг доверенный(1)/нет(0)
- `published_at` - дата публикации
- `created_at` - дата создания
- `alternate_url` - url вакансии
- `working_days` - дни работы (Работа только по сб и вс)
- `working_time_intervals` - время работы (Можно работать сменами по 4–6 часов в день)
- `working_time_modes` - особенности времени работы работы (Можно начинать работать после 16:00)
- `languages` - требуемые языки

Структура базы данных

Проектирование хранилища в Clickhouse для хранения данных из HH

```
CREATE TABLE default.hh_vacanies

(`id` BIGINT CODEC(ZSTD(1)), -- идентификатор вакансии

 `name` String CODEC(LZ4), -- название вакансии

 `premium` UInt8 CODEC(ZSTD(1)), -- флаг премиум вакансии

 `salary_from` UInt16 CODEC(ZSTD(1)), -- зарплата от

 `salary_to` UInt16 CODEC(ZSTD(1)), -- зарплата до

 `salary_currency` LowCardinality(String) CODEC(LZ4), -- зарплата валюта

 `salary_gross` UInt8 CODEC(ZSTD(1)), -- флаг до вычета(1)/чистыми(0)

 `vacancy_type` LowCardinality(String) CODEC(LZ4), -- тип вакансии (открытая, закрытая, архивная и т.д.)

 `experience` LowCardinality(String) CODEC(LZ4), -- требуемый опыт (нет опыта, от 1 до 2 лет и т.д.)

 `schedule` LowCardinality(String) CODEC(LZ4), -- график работы (Удаленная работа, Гибкий график и т.д.)

 `employment` LowCardinality(String) CODEC(LZ4), -- занятость (Частичная занятость, Полная занятость и т.д.)

 `description` String CODEC(ZSTD(1)), -- описание вакансии

 `key_skills` Array(String) CODEC(ZSTD(1)), -- ключевые навыки

 `professional_roles` LowCardinality(String) CODEC(LZ4), -- роль профессии

 `employer_id` BIGINT CODEC(ZSTD(1)), -- id нанимателя

 `employer_name` String CODEC(ZSTD(1)), -- название нанимателя

 `employer_url` String CODEC(ZSTD(1)), -- ссылка на профиль нанимателя

 `employer_logo_url` String CODEC(ZSTD(1)), -- ссылка на аватар нанимателя

 `employer_vacancies_url` String CODEC(ZSTD(1)), -- ссылка на вакансии нанимателя

 `trusted_employer` UInt8 CODEC(ZSTD(1)), -- флаг доверенный(1)/нет(0)

 `published_at` DateTime CODEC(DoubleDelta, ZSTD(1)), -- дата публикации

 `created_at` DateTime CODEC(DoubleDelta, ZSTD(1)), -- дата создания

 `alternate_url`  String CODEC(ZSTD(1)), -- url вакансии

 `working_days` LowCardinality(String) CODEC(LZ4), -- дни работы (Работа только по сб и вс)

 `working_time_intervals` LowCardinality(String) CODEC(LZ4), -- время работы (Можно работать сменами по 4–6 часов в день)

 `working_time_modes` LowCardinality(String) CODEC(LZ4), -- особенности времени работы работы (Можно начинать работать после 16:00)

 `languages` Array(LowCardinality(String)) CODEC(ZSTD(1)) -- требуемые языки

)

   ENGINE = MergeTree()

      ORDER BY (published_at, id)

      PARTITION BY toYYYYMM(published_at);
```

Выделим основные особенности данной таблицы:

- мы не разделяем сущность вакансии и сущность наниматель, так как в колоночных СУБД данные хранят обычно в больших таблицах из-за высокой стоимости в соединение таблиц при построение совмещенных запросов по нескольким сущностям;
- с той же целью в таблице есть поля с типом данных Array – languages, key skills;
- в качестве движка таблиц использован MergeTree, так как это самый функциональный движок. Основная идея, заложенная в основу движков семейства MergeTree следующая. Когда у вас есть огромное количество данных, которые должны быть вставлены в таблицу, вы должны быстро записать их по частям, а затем объединить части по некоторым правилам в фоновом режиме. Этот метод намного эффективнее, чем постоянная перезапись данных в хранилище при вставке. Этот движок отлично подходит для нашей задачи.
- партиционирование данных будем делать по полю published_at, причем только в разрезе месяца. Это позволит эффективно читать данные, если задан какой-либо временной промежуток, например, за последнюю неделю
- задан составной ключ сортировки, который состоит из полей published_at и id. Первичный ключ не задан явно, так как он берется автоматически равным ключу сортировки.

Проектирование хранилища в PostgreSQL для хранения данных из HH

```
create table if not exists vacancy
(
    id       varchar(15) primary key,
    name varchar(40) not null,
    billing_type varchar(40),


    sallary_from int,
```

```
    salary_to int,
    salary_currency varchar(10),
    salary_gross boolean,

    vacancy_type varchar(10),

    experience text,
    schedule varchar(20),

    employment text,
    description text,
    key_skills varchar(20)[],
    professional_roles text[],

    published_at text,
    created_at text,
    alternate_url text,
    working_days text[],
    working_time_intervals text[],
    working_time_modes text[],
    languages text[],
    area text,
    address text,
    branded_description text,
    accept_temporary boolean,

    employer_id int null

);

create table if not exists employer
(
    employer_id         varchar(20) primary key ,
    employer_name       varchar(128),
    employer_logo_url         varchar(40),
    employer_url     varchar(400) not null,
    employer_vacancies_url text,
```

```
    employer_trusted boolean not null
);
```

Загрузка данных в БД

Загрузка всей базы данных

1. **ClickHouse**

Данные загружали пачками со средним размером 230 – 240 Мб.

Загрузка происходила с помощью утилиты clickhouse-client.

Пример импорта:

```
clickhouse-client --format_csv_delimiter="|" --query "INSERT INTO default.hh_vacanies FORMAT CSV" < 0.csv
```

Среднее время выполнения запроса:

```
root@fb3c3a8358c3:/# time clickhouse-client --format_csv_delimiter="|" --query "INSERT INTO default.hh_vacanies FORMAT CSV" < 2.csv

real    0m1.184s
user    0m0.494s
sys     0m0.406s
```

Общее время полной загрузки данных в Clickhouse – 10,98 с

Загружаемый размер данных – 2,98 Гб

Размер таблицы после загрузки:

	table	size	rows	latest_modification	bytes_size	engine	primary_keys_size
1	default.hh_vacanies	398.49 MiB	773882	2023-04-14 14:56:40	417843642	MergeTree	4.27 KiB

Рис. 1 – Данные таблицы hh_vacancies в Clickhouse

2. **PostgreSQL**

Данные загружали пачками со средним размером 230 – 240 Мб.

Загрузка происходила через IDE DataGrip.

Пример импорта:

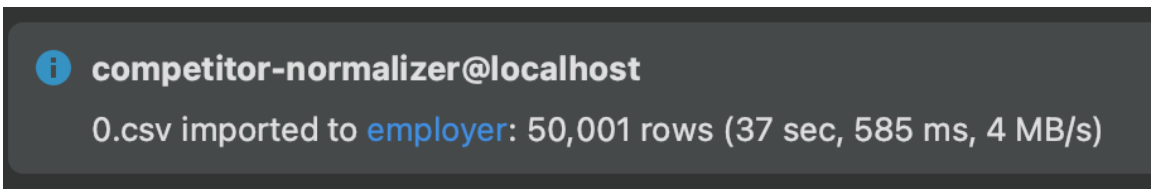
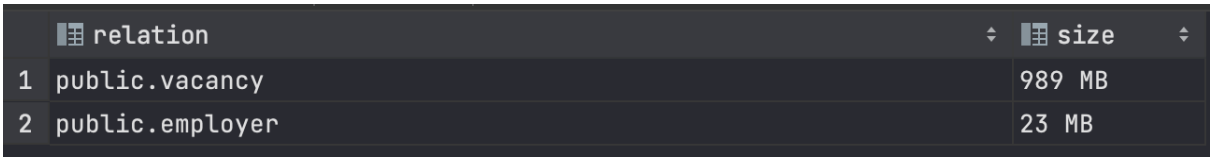


Рис.2 – Время загрузки данных (50000 строк) в PostgreSQL

Общее время полной загрузки данных в PostgreSQL - 4 min, 15 sec, 648 ms

Загружаемый размер данных – 2,98 Гб

Размер таблицы после загрузки:



	relation	size
1	public.vacancy	989 MB
2	public.employer	23 MB

Рис. 3 – Данные таблиц в PostgreSQL

Запрос для получения данных для размера таблиц:

```
SELECT nspname || '.' || relname AS "relation",
    pg_size_pretty(pg_relation_size(C.oid)) AS "size"
FROM pg_class C
     LEFT JOIN pg_namespace N ON (N.oid = C.relnamespace)
WHERE nspname NOT IN ('pg_catalog', 'information_schema')
ORDER BY pg_relation_size(C.oid) DESC
LIMIT 2;
```

3. Итог

	Clickhouse	**PostgreSQL**
Итоговый размер, Мб	398,5	1012 (989 + 23)
Время полной загрузки, с	10, 98	351,1 (255,65 + 95,5)

Таб.1 – Сравнительный анализ размеров базы данных и времени загрузки

Анализ времени загрузки от количества

1. ClickHouse

Размер/№	**1, мс**	**2, мс**	**3, мс**	**4, мс**	**5, мс**	**6, мс**	**7, мс**	**8, мс**	**9, мс**	**10, мс**	**Среднее**
13100	0470	0199	0196	0197	0207	0236	0231	0189	0184	0218	232
17500	0449	0283	0290	0284	0283	0257	0307	0255	0260	0351	301
26500	0527	0412	0503	0493	0440	0403	0573	0380	0384	0676	479
35000	0902	0566	0766	0688	0838	0825	0504	0702	0667	0627	708
52500	1175	0870	0728	1013	0987	1136	1194	1191	1162	1234	1069
78500	1947	1797	1680	1574	1494	1689	1491	1508	1653	1653	1648

125000	2237	2372	2205	2378	2431	2526	2352	2364	2529	2310	2370

Таб.2 – Сравнительный анализ времени загрузки данных в Clickhouse от размера файла

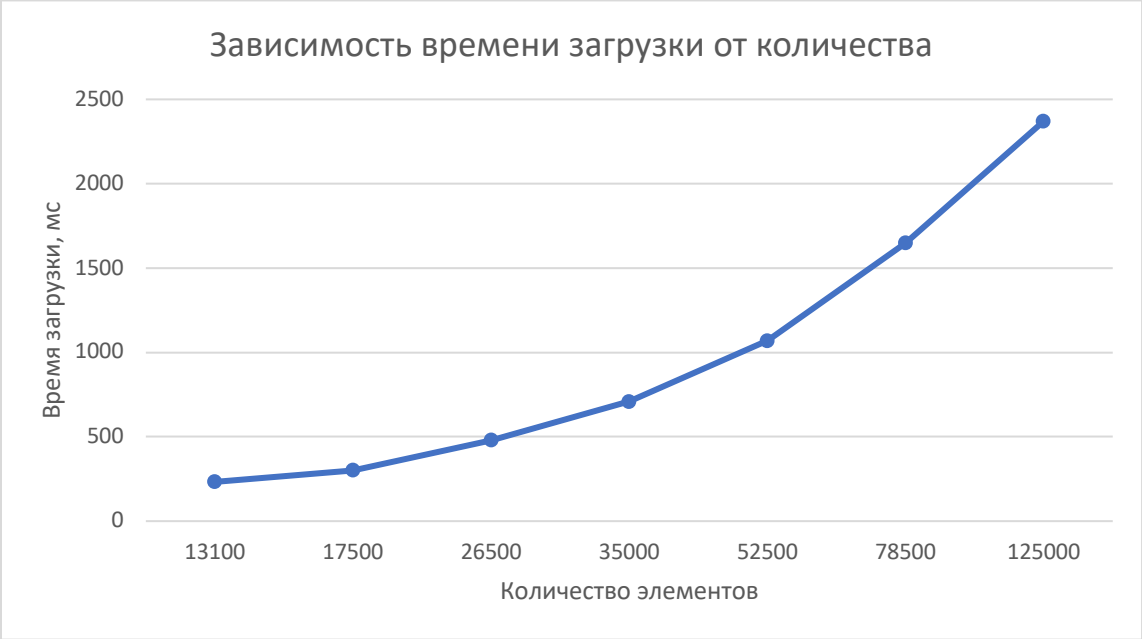


Рис.4 – Сравнительный анализ времени загрузки данных в Clickhouse от размера файла

2. PostgreSQL

Размер/№	**1, мс**	**2, мс**	**3, мс**	**4, мс**	**5, мс**	**6, мс**	**7, мс**	**8, мс**	**9, мс**	**10, мс**	**Среднее**
13100	5004	4657	4696	4921	4769	4652	4807	4591	5407	5299	4880
17500	7005	6884	6973	6911	6663	7181	7029	7068	6946	6907	6956
26500	10580	10553	10162	10250	10350	10215	10541	10175	10222	10273	10332
35000	13626	14923	14004	13434	13464	13434	13475	13442	13821	14348	13797
52500	22343	20419	20481	20569	20620	20636	21028	21129	21201	21147	20957
78500	30957	30880	30717	30707	30601	30877	30627	30616	30726	31534	30824
125000	49070	47898	48869	50451	48854	48324	48024	47958	48302	48924	48667

Таб.3 – Сравнительный анализ времени загрузки данных в PostgreSQL от размера файла

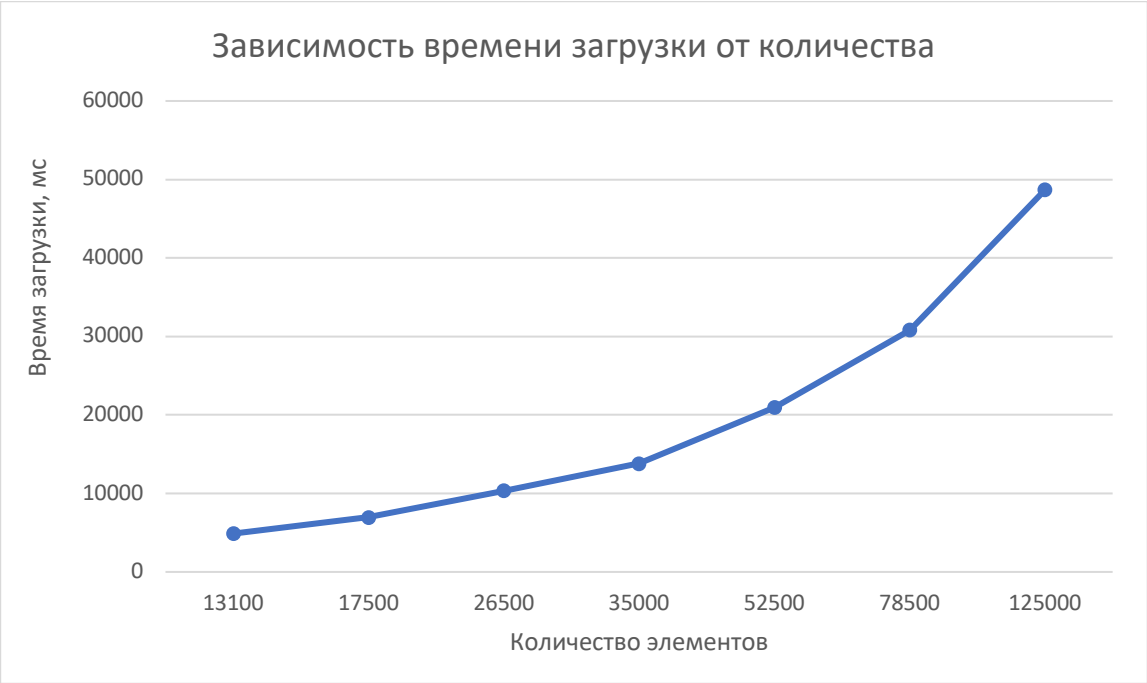


Рис.5 – Сравнительный анализ времени загрузки данных в PostgreSQL от размера файла

3. Итог

Размер/СУБД	**ClickHouse, мс**	**PostgreSQL, мс**
13100	232	4880
17500	301	6956
26500	479	10332
35000	708	13797
52500	1069	20957
78500	1648	30824
125000	2370	48667

Таб.4 – Сравнительный анализ времени загрузки от количества данных

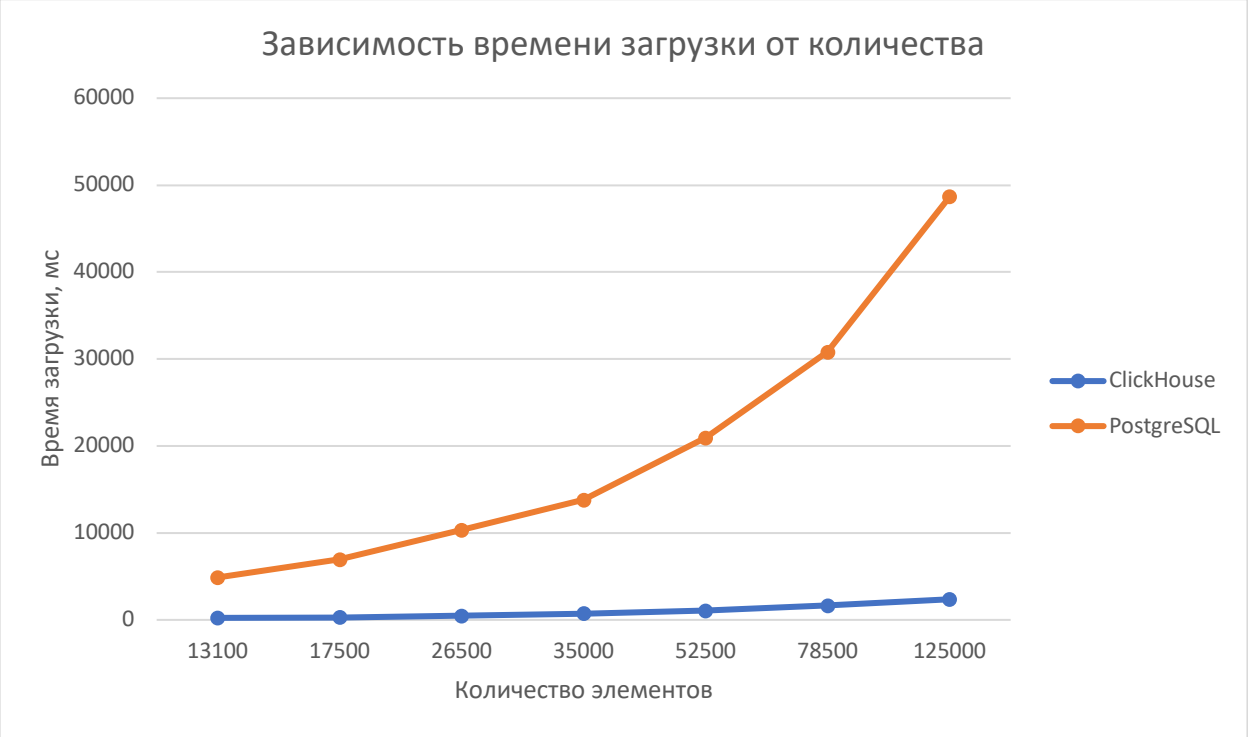


Рис.6 – Сравнительный анализ времени загрузки от количества данных

Анализ и сравнение работы СУБД

Сравнение скорости выполнения на примере типовых запросов

№ запроса	Время ClickHouse, мс	Время PostgreSQL, мс
1	111	2277
2	132	2004
3	154	4277
4	150	1959
5	137	1620

Таб.5 – Скорость выполнения типовых запросов

1. **Количество работодателей в каждом типе занятости**

Текст запроса Clickhouse:

```
select employment, count(distinct employer_id)
from hh_vacanies
group by employment
```

Текст запроса PostgreSQL:

```
select employment, count(distinct employer_id)
```

from vacancy
group by employment

Результат:

Тип занятости	**Количество работодателей**
Стажировка	50234
Проектная работа	50217
Волонтерство	50156
Частичная занятость	50072
Полная занятость	49728

Таб.6 – Количество работодателей в каждом типе занятости

Итог:

Время выполнение в Clickhouse – 111 мс

Время выполнения в PostgreSQL – 2277 мс

2. Количество работодателей по каждому языку

Текст запроса Clickhouse:

select arrayJoin(languages), count(id)
from hh_vacanies
group by 1

Текст запроса PostgreSQL:

select unnest(languages), count(id)
from vacancy
group by 1

Результат:

Язык	**Количество работодателей**
'RUS'	623879
'ENG'	311966

Таб.7 – Количество работодателей по каждому языку

Итог:

Время выполнение в Clickhouse – 132 мс

Время выполнения в PostgreSQL – 2004 мс

3. Название и количество вакансий по каждой компании (топ 10)

Текст запроса Clickhouse:

```
select employer_name, count(id)
from hh_vacanies
group by 1
order by 2 desc
limit 10
```

Текст запроса PostgreSQL:

```
select e.employer_name, count(vacancy.id)
from vacancy
left join employer e on vacancy.employer_id = e.employer_id
group by 1
order by 2 desc
limit 10
```

Результат:

Название	**Количество**
МАГНИТ, Розничная сеть	22155
Яндекс Крауд	5677
Пятёрочка	4792
Тинькофф	4112
СБЕР	3859
Яндекс	3428
Департамент Ф53	2509

Ростелеком	2434
Ozon	2022
Консалтинг Групп	1793

Таб.8 – Название и количество вакансий по каждой компании

Итог:

Время выполнение в Clickhouse – 154 мс

Время выполнения в PostgreSQL – 4277 мс

4. Средний потолок зарплат по профессиям (топ 10)

Текст запроса Clickhouse:

```
select arrayJoin(professional_roles), avg(salary_to)
from hh_vacanies
group by 1
having count(id) > 100
order by 2 desc
limit 10
```

Текст запроса PostgreSQL:

```
select unnest(professional_roles), avg(salary_to)
from vacancy
WHERE salary_currency = 'RUR'
group by 1
having count(id) > 100
order by 2 desc
limit 10
```

Результат:

Название	**Среднее значение потолка зарплаты**
'Агент по недвижимости'	336446.256851155293

'Брокер'	335817.5
'Руководитель группы разработки'	255518.339901477833
'DevOps-инженер'	233027.027027027027
'Коммерческий директор (CCO)'	232906.687022900763
исполнительный директор (CEO)'	219133.861224489796
'Генеральный директор	219133.861224489796
'Руководитель отдела продаж'	199754.453773584906
'Технический директор (CTO)'	193159.48275862069
'Руководитель строительного проекта'	188453.516441005803

Таб.9 – Средний потолок зарплат по профессиям

Итог:

Время выполнение в Clickhouse – 150 мс

Время выполнения в PostgreSQL – 1959 мс

5. Среднее значение зарплат в регионах (топ 10)

Текст запроса Clickhouse:

```
select area, avg(salary_to)
from hh_vacanies
group by 1
having count(id) > 100
order by 2 desc
limit 10
```

Текст запроса PostgreSQL:

```
select area, avg(salary_to)
from vacancy
WHERE salary_currency = 'RUR'
group by 1
having count(id) > 100
```

order by 2 desc
limit 10

Результат:

Название	**Среднее значение потолка зарплаты**
Адлер	206620.883534136546
Сочи	182900.4140625
Норильск	120916.412371134021
Новый Уренгой	113778.241452991453
Москва	112948.047428356902
Магадан	107396.529850746269
Ноябрьск	98466.880681818182
Якутск	96112.838050314465
Щербинка	95283.380530973451
Нижневартовск	92566.093922651934

Таб.10 – Среднее значение зарплат в регионах

Итог:

Время выполнение в Clickhouse – 137 мс

Время выполнения в PostgreSQL – 1620 мс

Анализ времени выполнения запросов

Будем рассматривать на основе запроса 3. Каждый эксперимент был повторен 10 раз для нахождения среднего времени и предотвращения влияния выбросов.

Кол-во строк	**Время выполнения в PostgreSQL, мс**	**Время выполнения в Clickhouse, мс**
50	196,3	22,2
100	214,4	25,7
200	358,1	41,3

350	880,8	53,1
500	1317,7	76,9
672	2615,6	85,4

Таб.10 – Зависимость времени выполнения от количества строк

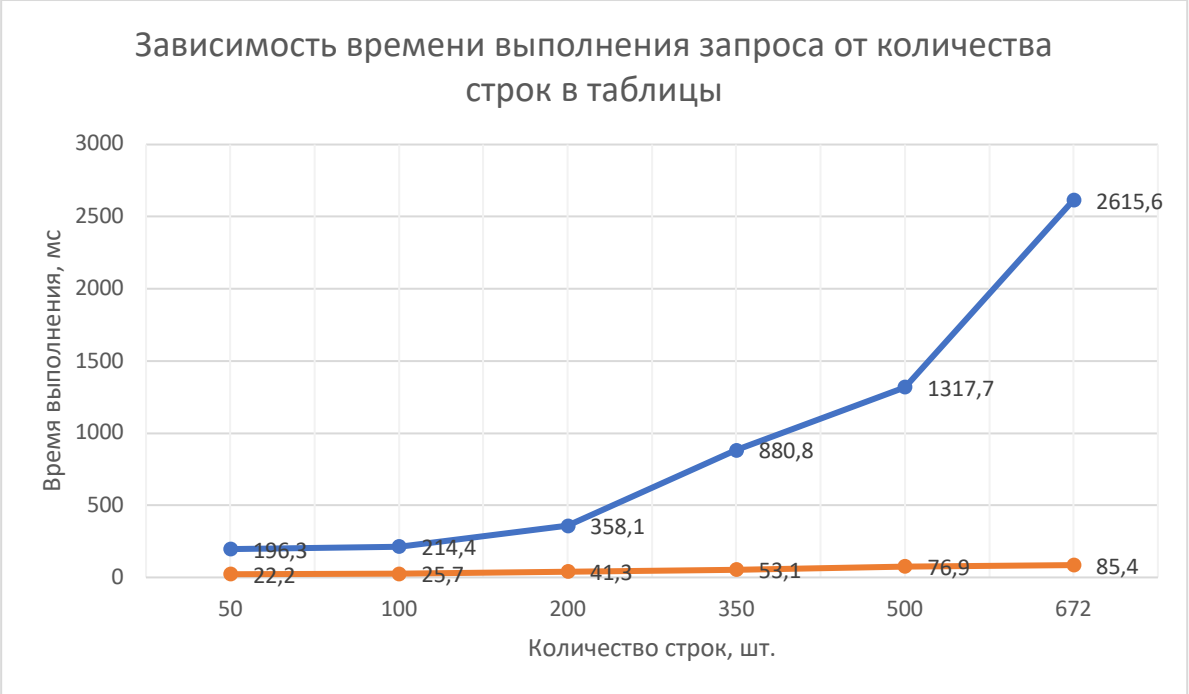


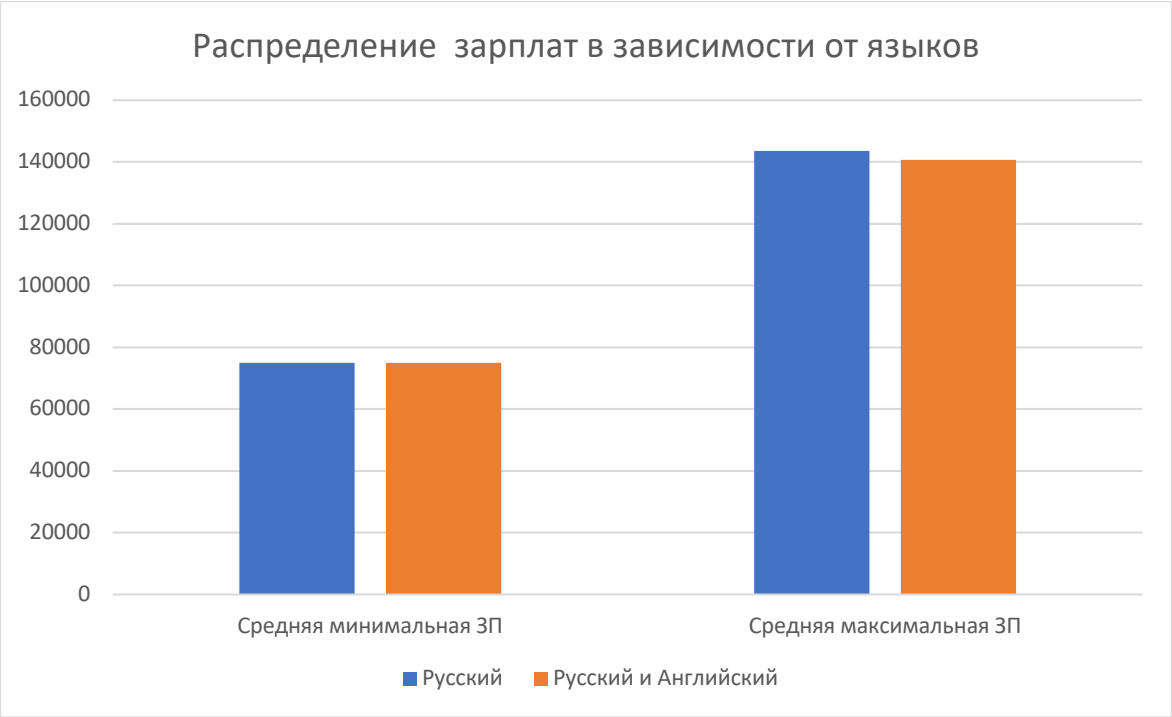
Рис.6 – Зависимость времени выполнения запроса от количества строк

Аналитические запросы

Зависимость ср. з/п от количества языков

Языки	Ср. мин. з/п	Ср. макс. з/п
['RUS']	74988.08240944712	143557.11431080624
['RUS', 'ENG']	75037.23485171405	140754.15646594274

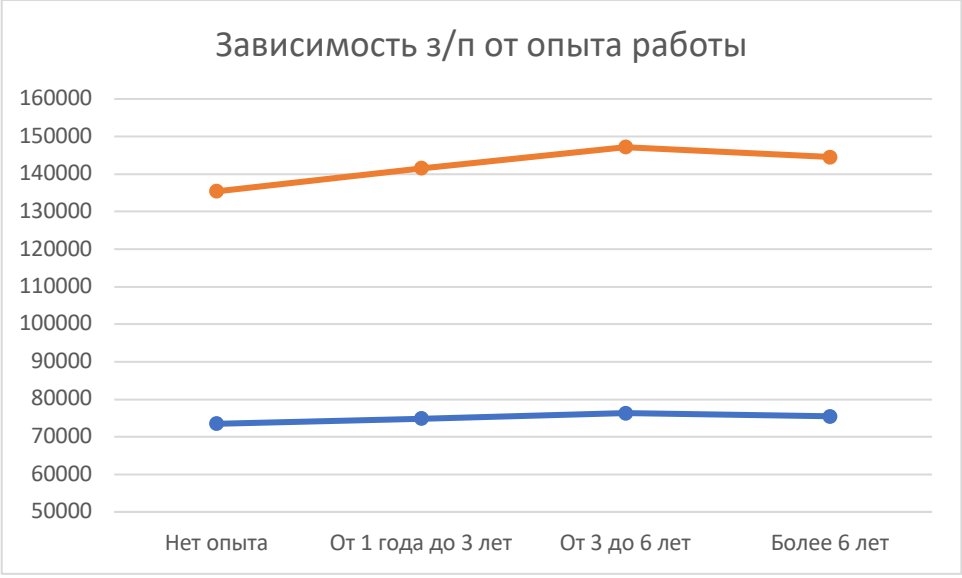
Таб.11 – Зависимость ср. з/п от количества языков



Зависимость з/п от опыта работы

Опыт	Ср. мин. з/п	Ср. макс. з/п
Нет опыта	73474.0247584541	135382.9846657003
От 1 года до 3 лет	74829.48181373818	141561.01037194725
От 3 до 6 лет	76329.15062927575	147148.47397533443
Более 6 лет	75406.11105018614	144471.72499585338

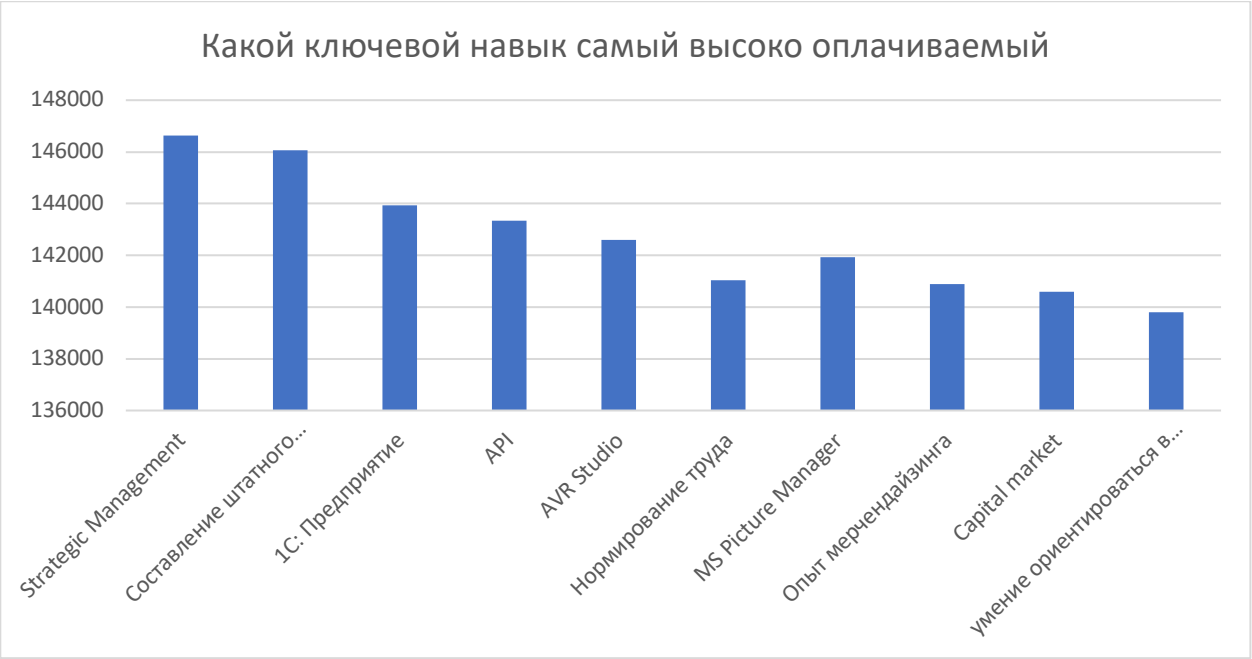
Таб.12 – Зависимость з/п от опыта работы



Какой ключевой навык самый высокооплачиваемый

Навык	Ср. з/п
Strategic Management	146645.12135420076
Составление штатного расписания, коллективного договора, положения по премированию	146064.2595545564
1С: Предприятие	143946.65133864438
API	143440.52473095755
AVR Studio	142586.58846344065
Нормирование труда	141217.818208013
MS Picture Manager	141040.84541295137
ОПЫТ МЕРЧЕНДАЙЗИНГА	140927.76136185142
Capital Market	140686.57391156728
умею ориентироваться в нестандартных ситуациях, обращаться с конфиденциальной информацией,	135251.4164956539

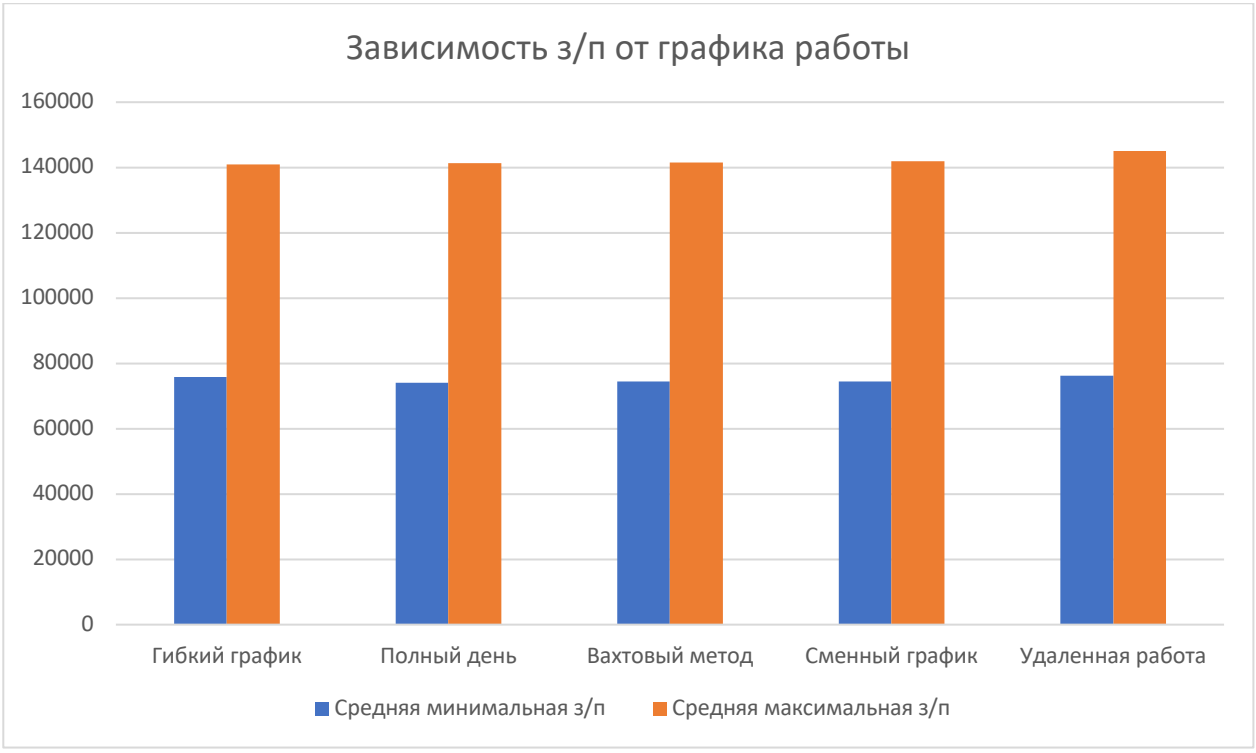
Таб.13 – Какой ключевой навык самый высокооплачиваемый



Зависимость з/п от графика работы

График	**Ср. мин. з/п**	**Ср. макс. з/п**
Гибкий график	75842.98992766727	141019.02493111923
Полный день	74060.82956339719	141235.42010588286
Вахтовый метод	74485.92428536504	141470.41353091435
Сменный график	74511.02999350321	141960.43600748485
Удаленная работа	76160.1795987104	145073.43555378626

Таб.14 – Зависимость з/п от графика работы


Зависимость з/п от графика работы
160000
140000
120000
100000
80000
60000
40000
20000
0
Гибкий график
Полный день
Вахтовый метод
Сменный график
Удаленная работа
Средняя минимальная з/п
Средняя максимальная з/п

Заключение

По итогам у нас получилось, что база данных ClickHouse лучше подходит для выполнения аналитических запросов за счет лучших алгоритмов сжатия данных. Она может быстрее чем PostgreSQL импортировать данные из файлов в формате csv. Разница во времени импортирования более 10 раз. Так же был проведен сравнительный анализ выполнения типовых запросов. Во всех случаях ClickHouse оказался быстрее чем PostgreSQL.

Выводы по аналитическим запросам:

- Средняя минимальная з/п больше у вакансий, в которых требуется только русский язык. Это обусловлено тем, что вакансии требующие знаний двух языков более специфичны и соответственно оплачиваются выше.
- Наблюдается прямо пропорциональная зависимость размера з/п от опыта работы: чем больше опыт, тем больше з/п. Это не соответствует только при опыте более 6 лет, так как эти вакансии соответствуют руководящим должностям и поэтому в них з/п считается иначе.
- Самыми высокооплачиваемыми навыками являются навыки руководящих должностей, такие как **Strategic Management** и составление штатных расписаний. Также в топе навыки IT специальностей.
- Удаленная работа имеет самое большое значение з/п, так как по большей части гибкий график имеют работники IT специальностей, которые имеют большие з/п.

Список использованных источников

1. Документация PostgreSQL https://www.postgresql.org/
2. Документация ClickHouse https://clickhouse.com/
3. Документация HeadHunter API https://dev.hh.ru/