

УДК 004.65

## ИССЛЕДОВАНИЕ ПРЕИМУЩЕСТВ НЕРЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ REDIS

Ремеслов В.Г., студент гр.Ист-221, II курс  
Семенова О.С., к.т.н., доцент  
Кузбасский государственный технический университет  
имени Т.Ф.Горбачева,  
г. Кемерово

Современные базы данных (БД) NoSQL – это системы управления информацией, которые выделяются на фоне классических реляционных систем за счет передовых решений, применяемых для хранения и обработки информации. К таким решениям относят:

- гибкость структуры, достигаемая за счет отсутствия жестких правил организации данных;
- масштабирование, достигаемое за счет возможности добавления новых узлов хранения данных;
- высокая скорость обработки запросов пользователей за счет хранения данных в оперативной памяти;
- поддержка хранения неструктурированных данных, таких как результаты выполнения запросов, текстовые документы, видео, изображения и др.

Применение NoSQL баз данных наиболее актуально при манипуляциях с огромными массивами разнородных данных, а также в сценариях, где требуется распределенная обработка данных и надежное хранение информации.

Существует большое количество нереляционных баз данных, но самыми распространенными являются MongoDB, Cassandra, Redis, Amazon DynamoDB и Neo4j. Они используются в таких областях, как аналитика больших данных, мобильные приложения, интернет-магазины, игровые платформы.

Рассмотрим преимущества нереляционных БД на примере БД Redis, работающей по принципу "ключ-значение" и предназначенной для сверхбыстрого извлечения хранимой информации. Отличительными особенностями Redis являются простота использования, высокая производительность, поддержка транзакций и репликации, возможность кэширования данных и обработки очередей задач.

Для примера возьмем реляционную систему управления базами данных (СУБД) PostgreSQL и разместим в ней БД, состоящую из 7 взаимосвязанных таблиц (рисунок 1). Поместим в таблицы тестовые данные (самая большая таблица public.data\_scientists\_salary\_finish содержит 7506 записей, самая маленькая public.company\_size – 3 записи). Создадим запросы к данным, нахо-

дящимся в реляционной БД, и сохраним результаты выполнения запросов в нереляционной БД Redis. Сравним время, затрачиваемое на извлечение данных из реляционной БД (с помощью классических SQL-запросов), и время, затрачиваемое на извлечение данных из нереляционной БД (с помощью операторов Redis).

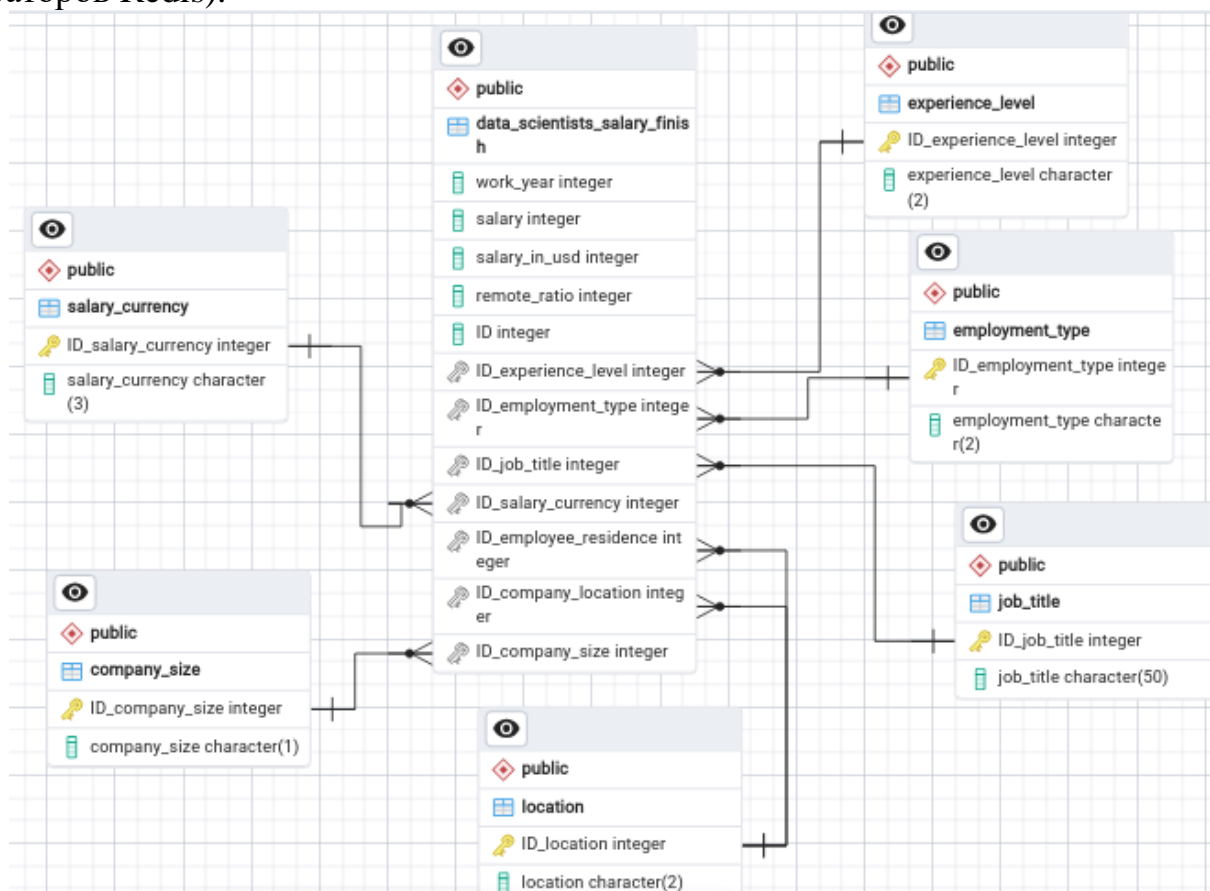


Рисунок 1 – Диаграмма БД

Возьмем затратный запрос, возвращающий 7506 записей из 6 взаимосвязанных таблиц:

```

SELECT work_year, salary, company_size, employment_type, experience_level,
job_title, location FROM public.data_scientists_salary_finish as ds
INNER JOIN company_size on
ds."ID_company_size"=company_size."ID_company_size"
INNER JOIN employment_type on
ds."ID_employment_type"=employment_type."ID_employment_type"
INNER JOIN experience_level on
ds."ID_experience_level"=experience_level."ID_experience_level"
INNER JOIN job_title on ds."ID_job_title"=job_title."ID_job_title"
INNER JOIN location on ds."ID_location"=location."ID_location"
    
```

Время выполнения SQL-запроса – 0.063 с.

С помощью метода `rpush <ключ> <значение>` поместим результат выполнения SQL-запроса в Redis, а потом его извлечем с помощью метода `mget`

<ключ>. В качестве ключа будет выступать сама SQL-инструкция, а в качестве значения – результат выполнения запроса.

В результате получено время извлечения сохраненного результата SQL-запроса из Redis – 0.0061 с., что 10,5 раз быстрее, чем непосредственное выполнение самого запроса.

Возьмем менее затратный запрос, возвращающий также 7506 записей, но только из одной таблицы (затратная операция INNER JOIN отсутствует):  
SELECT work\_year, salary FROM public.data\_scientists\_salary\_finish

Время выполнения SQL-запроса – 0.011 с., время извлечения сохраненного результата SQL-запроса из Redis – 0.0059 с., что 1,5 раз быстрее, чем непосредственное выполнение самого запроса.

Таким образом, наибольший выигрыш по времени доступа к данным из Redis получаем для сложных запросов, наименьший – для простых. Причем, как видно из примера, время извлечения данных из Redis практически не зависит от размера извлекаемых данных.

Следует заметить, что несмотря на свои многочисленные преимущества, у Redis есть некоторые недостатки, которые стоит учитывать при выборе этой базы данных для конкретного проекта. К недостаткам следует отнести ограничение на объем хранимых данных, относительную сложность масштабирования, ограниченную функциональность, отсутствие поддержки полноценных транзакций.

Однако, если требуется высокая производительность и быстрый доступ к данным, то нереляционная СУБД Redis окажется незаменимой.

### **Список литературы:**

1. Эрик Редмонд, Джим Р. Уилсон Семь баз данных за семь недель. Введение в современные базы данных и идеологию NoSQL [Текст] / Эрик Редмонд, Джим Р. Уилсон : ДМК Пресс, 2015. – 384 с.
2. Redis. Режим доступа: <https://redis.io/>