Технические науки

УДК 004.652.4; 004.22

**Царегородцев Дмитрий Александрович**

*магистрант кафедры информационных систем и технологий*

*Казанского федерального университета*

**Tsaregorodtsev Dmitry**

*Master Student of the Department of Information System and Technology*

*Kazan Federal University*

# ВЕРТИКАЛЬНАЯ НЕЛИНЕЙНАЯ СХЕМА ХРАНЕНИЯ ДАННЫХ В КОРПОРАТИВНЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ
# VERTICAL NONLINEAR DATA STORAGE SCHEME IN CORPORATE INFORMATION SYSTEMS

***Аннотация.*** *Разработана гибкая схема хранения данных для мастер-справочником корпоративных информационных систем. Предусмотрена основа функционала для постобработки данных*

***Ключевые слова:*** *архитектура корпоративных систем, хранение данных, базы данных.*

***Summary.*** *A flexible data storage scheme has been developed for the master directory of corporate information systems. Provides a basis for post-processing functionality*

***Key words:*** *architecture of corporate systems, data storage, database.*

**Вступление.** Корпоративная информационная система – это система, которая нацелена на автоматизацию процессов при ведении многих видов деятельности внутри компании [1]. Несмотря на то, что корпоративные системы не обладают такими же темпами устаревания, как системы, нацеленные на продажу и максимально частую итерацию обновления, они

все же требуют масштабирования и развития. Чем лучше система будет отвечать на требования компании к автоматизации, тем выше будут достигнуты результаты.

Чаще всего информационная система подразумевает собой сеть взаимосвязанных подсистем, сервисов, микро-сервисов, которые обмениваются данными с помощью интеграционных решений [2]. В ряде звеньев этой цепи используются мастер-справочники. Мастер-справочник – это система, которая является центральным хранилищем в какой-то предметной области. Другие подсистемы, либо сервисы, используют ту часть данных справочника, которая необходима для их работы. Зависимые подсистемы используют либо данные напрямую из источника, либо используют кеш на своей стороне, либо пользуются помощью сторонних решений в получении данных из справочника. Сторонним решением может являться отдельная база данных, которая содержит срез основных нечувствительных данных, по которым поступают запросы чаще всего, либо BI-решения, основанные на отчетах по данным [3, с. 196-199]. Примером мастер-справочника может служить система, которая хранит в себе всю информацию о сотрудниках. Грамотное проектирование и организация структуры хранения данных является очень важной задачи.

**Проблематика темы и решения поставленных задач.** Мастер-справочник хранит в себе данные о каком-то ключевом предмете, причем ключевых предметов может быть несколько. Обозначим ключевой предмет, как сущность. Сущность должна обладать набором свойств и значений, которые ее характеризуют. Чтобы реализовать хранение таких данных можно воспользоваться линейным отображением [4]. В качестве хранилища данных будем рассматривать реляционную базу данных, а в качестве парадигмы на уровне исходного кода – объектно-ориентированное программирование. Линейное отображение подразумевает, что каждая сущность будет представлена в виде отдельной

таблицы, а свойство сущности – колонкой в таблице. Соответственно на уровне программного кода каждая сущность – это класс, где свойства – поля класса. Данный подход не отличается гибкостью потому, что при заведении новых свойств, необходимо добавлять новый столбец в таблице, вносить новое поле в классе, что при нарастающем количестве свойств раздует соответствующие таблицы и классы. Последствия негативно отразятся на поддержке и сопровождении проекта, что неминуемо приводит к более скорой гибели системы в будущем. Линейное представление данных также ограничивает функциональные возможности системы

Гибким решением данного вопроса является нелинейное вертикальное хранение [5]. Вместо хранения всей информации в одной таблице по сущности, выносим в несколько таблиц. В первую очередь необходимо разделить сущность от ее свойств. Для этого будет заведено несколько таблицы. Первая таблица отражает типы свойств. Данная таблица служит для определения типа данных на уровне языка программирования, который будет работать с этими данными. Во второй таблице будут хранится метаданные свойства. К метаданным свойства относится ссылка на тип сущности, на тип данных из первой таблицы, системное имя. Здесь же мы можем указать данные для валидации данных при их обработке (регулярные выражения, методы, вызываемые через рефлексию в исходном программном коде), а также методы постобработки данных. На уровне мета-свойств появилась возможность разграничить права на редактирование данных на уровне отдельных значений, путем ссылки на доступные для данного поля роли. Такая функциональность была недоступна при линейном отображении. Следующей таблицей будут представленные значения свойств. В данной таблице будет ссылаться на мета-свойство, сущность, значение которой отображено, а также само значения. Данную таблицу можно расширить полями начала и окончания

действия значения. Таким образом появляется возможность отслеживать изменения конкретных полей [6]. Историчность данных была также недоступна при линейном ведении данных. Названия свойств, доступное для просмотра в клиентском приложении, также следует вынести в отдельную таблицу, в которой можно обеспечить поддержку нескольких языков. В таблице самой сущности останется лишь уникальный идентификатор, тип сущности и системная информация. В качестве типа сущности может выступать ссылка на таблицу типов, либо целочисленные значения. Системную информацию справедливо вести во всех таблицах, чтобы фиксировать дату и время изменения данных, а также каким пользователем она была произведена [7]. На рисунке 1 представлена диаграмма сущностей.

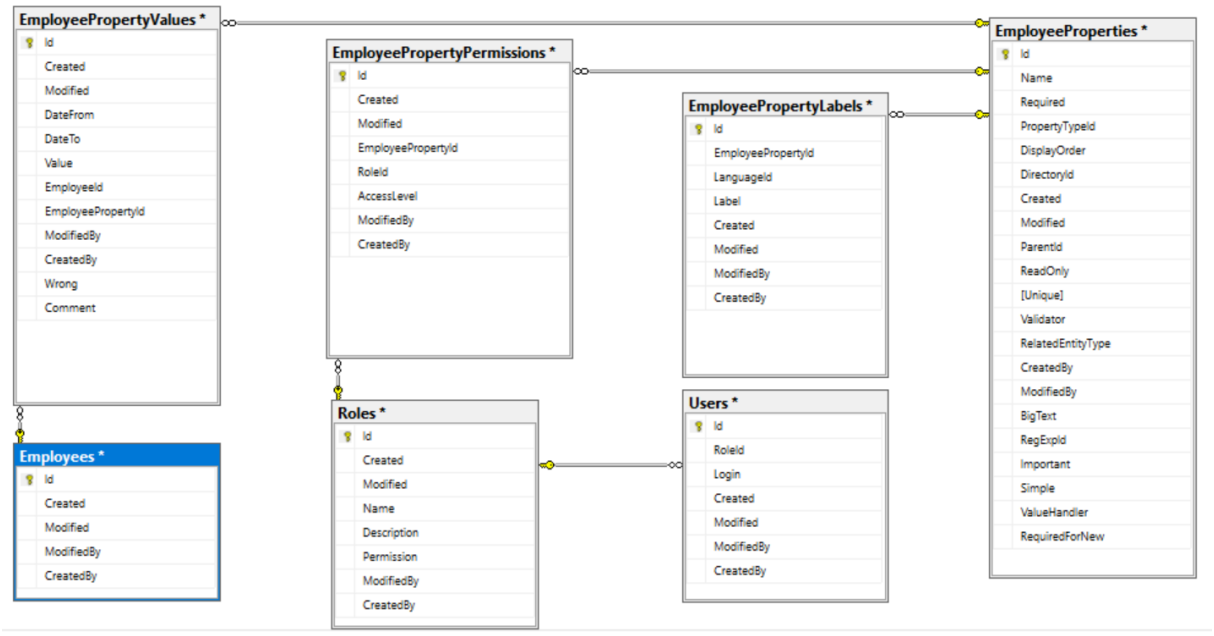


**Рис. 1. Диаграмма сущностей**

**Анализ решения.** Быстродействие выбранного решения имеет немаловажную роль. Проведем эксперимент на примере компании с численностью от 100 человек до 10000. Также будем использовать разное количество свойств: от 10 до 100. Фиксируем среднее время среди 100 запросов. Исследуем время, которое требуется для запроса поиска

сотрудника по фамилии. Тестовая машина, на которой производится исследование, имеет 12 ГБ оперативной памяти и двухъядерный процессор с таковой частотой 2.4 ГГц.

На рисунке 1 продемонстрирована зависимость времени выполнения запроса от количества записей в таблице значений. В таблице 1 подробно описаны различные комбинации количества сотрудников и количества свойств.

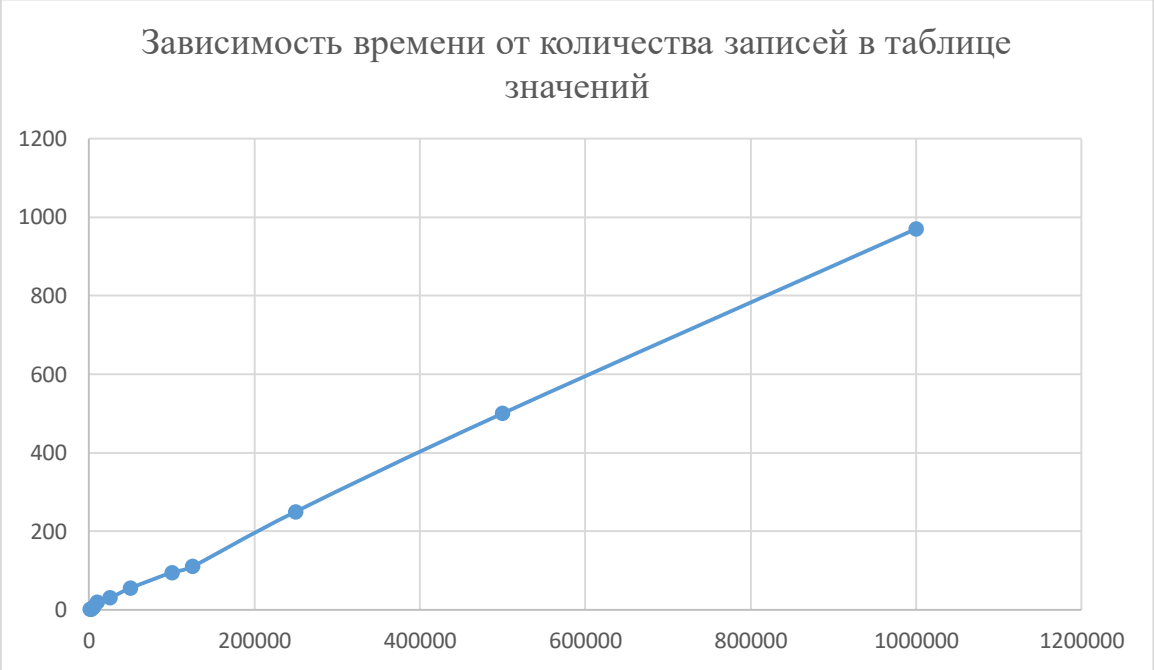


**Рис. 2. График зависимость времени от количества строчек в запросе**

*Таблица 1*

**Зависимость времени выполнения запроса от количества сущностей и свойств**

| Сущности / Свойства | **100** | **1000** | **5000** | **10000** |
|---|---|---|---|---|
| **10** | 2мс | 20мс | 55мс | 100мс |
| **25** | 2мс | 30мс | 110мс | 250 мс |
| **50** | 7мс | 55мс | 250мс | 500мс |
| **100** | 20 мс | 90мс | 500мс | 970мс |

**Выводы.** На основе проведенного эксперимента можно сделать вывод, что для малых или средних компаний предложенная структура данных отвечает на запросы достаточно быстро. Для крупных компаний время одного запроса может достигать 1 секунды. Однако крупные компании могут позволить повысить производительность покупкой более дорогостоящего оборудования, что должно значительно снизить время запроса.

## Литература

1. Фаулер М., Райс Д., Фоммел М., Хайэт Э., Ми Р., Стаффорд Р. Шаблоны корпоративных приложений. 4-е изд.: Пер. с англ. М.: Вильямс. 2016. 544 с.
2. Connolly Thomas, Begg Carolyn. Database Systems: A Practical Approach to Design, Implementation, and Management, 6th edition. M.: «Pearson». 2015. 1440 с.
3. Плещев В.В., Старк Т. Автоматизированные адаптивные методические системы обучения в области разработки корпоративных информационных систем // BI-технологии и корпоративные информационные системы в оптимизации бизнес-процессов. 2016. С. 196-199.
4. Бейли Линн Изучаем SQL. 3-е изд.: Пер. с англ. М.: Питер. 2012. 592 с.
5. Кэмпбелл Лейн, Мейджорс Черити «Базы данных». 1-е изд.: Пер с англ. М.: Питер. 2020. 304 с.
6. Осипов Д.Л. Технологии проектирования баз данных. М.: ДМК Пресс. 2019. 498 с.
7. Джеффри У., Дженнифер У. «Системы баз данных», 5-е изд.: Пер с англ. М.: Вильямс. 2017. 1088 с.