

ОСОБЕННОСТИ ХРАНЕНИЯ ДАННЫХ В РЕЛЯЦИОННЫХ СУБД

Кузмин И.А.

Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь

Научный руководитель: Пискун Г.А. – канд.техн.наук, доцент, доцент кафедры ПИКС

Аннотация. В статье рассмотрены основные принципы хранения данных при работе с СУБД. Показаны некоторые особенности хранения данных в различных СУБД.

Ключевые слова: БД, Базы данных, СУБД.

Введение. Одним из важнейших критериев для баз данных являются производительность, надежность и простота управления. Знание того, как большинство СУБД физически хранят данные во внешней памяти, параметры этого хранения и методы доступа, может помочь в создании баз данных с заданной производительностью.

Организация хранения. Основными единицами физического хранения являются блоки данных, экстенды и файлы (или разделы жесткого диска). Логический уровень включает пространства (или табличные пространства). Блок данных (или страница) является единицей обмена с внешней памятью. Размер страницы фиксирован для базы данных Oracle, а для других СУБД (DB2, Informix, Sybase) он устанавливается при создании. Очень важно правильно выбрать размер блока с самого начала, так как изменение размера в работающей базе практически невозможно (для этого часто проводятся испытания баз данных-прототипов).

Функции среды хранения базы данных. Базы данных отличаются от других хранилищ данных тем, что они обрабатывают информацию с помощью электронных носителей и систематизируют ее в соответствии с определенными правилами. Базы данных играют важную роль в обеспечении бесперебойной работы систем, которые содержат глобальную информацию, например, о государственных ресурсах и территориях. В них содержится подробная информация о строениях, гидрографии и растительности на этих территориях. Например, о государственных ресурсах или определенных территориях. В них детально расписаны все данные о строениях, гидрографии и растительности данных мест [1].

Механизмы среды хранения базы данных в СУБД управляют двумя группами ресурсов системы – ресурсами хранимых данных и ресурсами пространства памяти среды хранения. Они обеспечивают эффективное хранение и поддержку структуры хранимых данных, управление пространством памяти базы данных и отображение структуры хранимых данных в памяти, чтобы определить место их размещения при запоминании и поиске данных.

Важно отметить, что структура хранимых данных выбирается на основе эффективности хранения и доступа к данным, и может отличаться от структуры данных, которую видит пользователь. В графовых моделях данных используются сложные структуры хранимых данных и управление ими, в то время как реляционные системы используют более простые структуры. Будем далее условно называть хранимой записью единицу хранимых данных, которая используется в качестве аргумента в операциях над данными в среде хранения. Эти операции – запомнить новую запись, удалить запись, перезаписать существующую запись, найти запись.

Пространство памяти. Администратор выделяет пространство для базы данных на внешних устройствах, используя большие фрагменты: файлы и разделы диска. В первом случае доступ к диску осуществляется операционной системой, что имеет определенные преимущества, такие как возможность работы с файлами средствами ОС. Во втором случае сервер работает непосредственно с внешним устройством, что обеспечивает более высокую производительность. Использование дисков необходимо в случае, если кэш ОС не может работать в режиме сквозной (write-through) записи. Диски особенно полезны для ускорения

операций записи данных (подобный механизм поддерживается Oracle, DB2 и Informix; например, в DB2 данная единица размещения называется контейнером) [1].

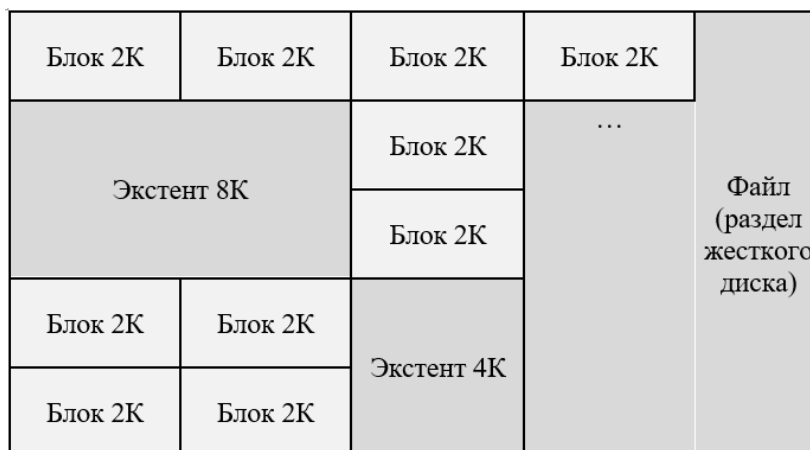


Рисунок 2 – Блоки, экстенды и файлы БД [1]

Поиск информации в базах данных. Важной методологической проблемой прикладной компьютерной лингвистики является правильная оценка необходимого соотношения между декларативной и процедурной компонентами систем автоматической обработки текстовой информации. Чему отдать предпочтение: мощным вычислительным процедурам, опирающимся на относительно небольшие словарные системы с богатой грамматической и семантической информацией, или мощной декларативной компоненте при относительно простых процедурных средствах. И высказали мнение, что второй путь предпочтительнее. Можно представить себе крайний случай такого предпочтения, когда лингвистическая задача решается путем создания большого словаря и поиска в этом словаре необходимой информации. Однако более реальна ситуация, когда задача решается путем последовательного поиска в небольшом количестве достаточно представительных словарей и выполнения несложных процедур преобразования результатов поиска.

Физическое хранение данных. Основные моменты в физическом хранении данных в СУБД (для примера выберем СУБД Oracle и PostgreSQL):

- обе СУБД используют сегменты и блоки для хранения данных, но Oracle помещает их в файлах данных, в то время как PostgreSQL использует индивидуальные файлы;

- обе СУБД поддерживают экстенды, причем PostgreSQL 9.6 может выделять место экстендами;

- обе СУБД поддерживают блоки;

- в Oracle размеры блоков, экстендов и сегментов настраиваемы, в то время как в PostgreSQL они вкомпилированы.

Структура блока данных:

- блок имеет заголовок со служебной информацией и таблицу указателей на записи;

- накладные расходы в Oracle составляют “on average data block overhead total 84 to 107 bytes” включая таблицу указателей на записи;

- накладные расходы в PostgreSQL составляют 24 байта заголовка +4*N где N – число записей в блоке;

- наполнение блока ведётся с конца;

- в Oracle в блоке содержится список транзакций, ждущих изменений данных в этом блоке;

- в Oracle блок может содержать записи отразных таблиц, в отличии от Postgres.

Ниже предоставлен изображение физического хранения данных в Oracle и в Postgres.

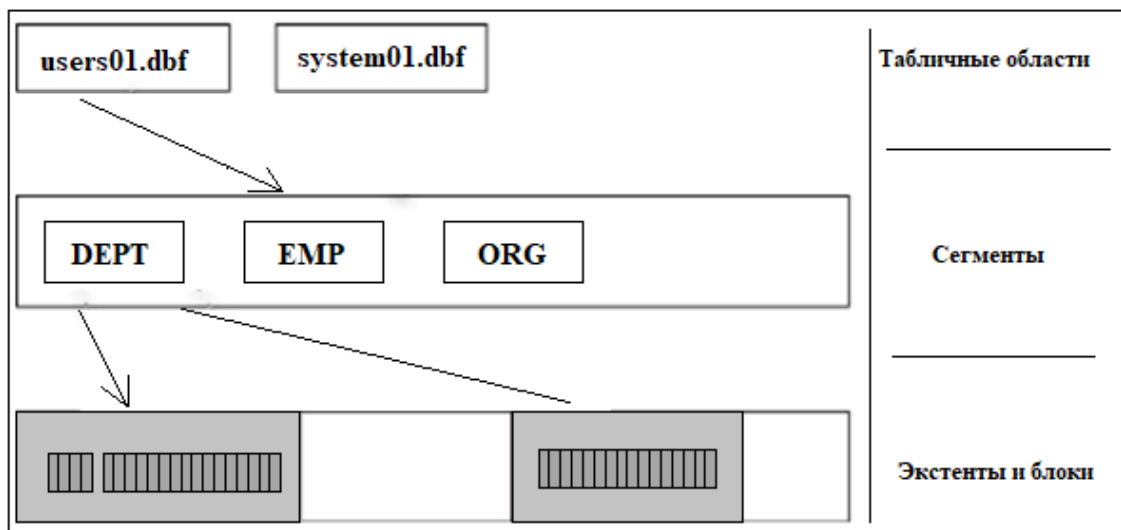


Рисунок 2 – физическое хранение данных в ORACLE [1]

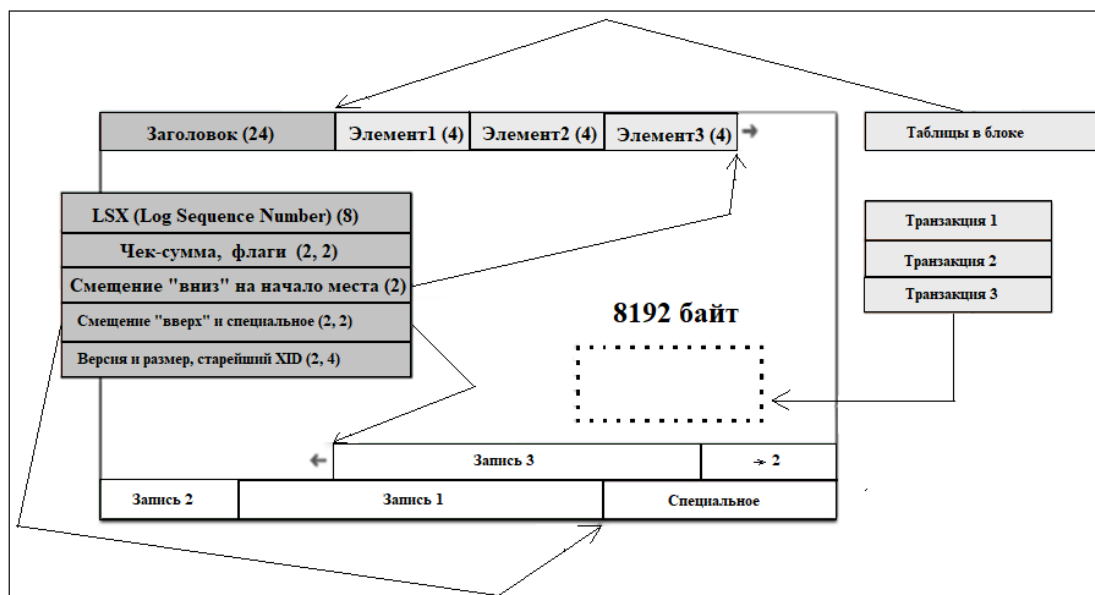


Рисунок 4 – физическое хранение данных в PostgreSQL [2]

Структура записей в PostgreSQL:

– заголовок записи в PostgreSQL составляет 23 байта + необязательная NULL-маска (битовая);

– в Oracle заголовок записи гораздо меньше, 3 байта в оптимальном случае;

– в Oracle отсутствует NULL-маска.

Основные моменты в версии и длинных значениях:

– PostgreSQL обеспечивает версию на уровне записи что напрямую влияет на размер заголовка записи;

– в PostgreSQL запись не может выходить за пределы блока, большие значения хранятся в специальных TOAST-таблицах;

– в Oracle длинные записи формируют цепочки из нескольких блоков.

Различия в организации сессий:

– каждое клиентское соединение обслуживается отдельным серверным процессом;

– Oracle позволяет настроить shared servers, для обслуживания многих соединений одним процессом;

– для Postgres рекомендуется использовать pgbouncer.

Единица данных. В базах данных каждая логическая единица данных (таблица или индекс) размещается только в одном пространстве, которое может быть распределено на несколько физических устройств или файлов. Физическое разделение может происходить не только между логическими единицами данных, такими как таблицы и индексы, но и в пределах одной логической структуры, такой как таблица, которая может храниться на нескольких дисках. Это называется горизонтальной фрагментацией, а в Oracle же используется термин «секционирование». Фрагментация является одним из способов повышения производительности баз данных.

Данные в базах данных хранятся в двоичном виде, как и любые другие данные, обрабатываемые компьютером. От данного способа представления данных пока что уйти не получится. Однако в каждой из двух представленных баз данных есть возможность представить данные в понятной для пользователя или программы форме (которая все же делает данные понятные человеку).

В большинстве БД текстовую (строковую информацию) можно хранить напрямую, без каких-либо модификаций, потому что БД имеют встроенные типы и алгоритмы для этих целей.

В большинстве БД текстовые данные можно хранить без изменений, используя встроенные типы данных и алгоритмы. Независимо от того, как данные хранятся, необходимо обеспечить их взаимосвязь с другими данными. В объектно-реляционных СУБД эта связь устанавливается через отношения, используя операции и законы теории множеств. В графовых базах данных связи между данными представляются в виде ребер графов или гиперграфов [4].

Заключение. Распределение данных в таблицах и стратегия работы оптимизатора СУБД, а также возможности диалекта SQL, влияют на эффективность использования методов доступа. Поэтому представленные рекомендации имеют общий характер и могут меняться в зависимости от конкретной ситуации. На этапах физического проектирования и настройки приходится искать компромисс между требуемыми характеристиками, которые могут противоречить друг другу. Использование индекса ускоряет обработку запросов, но требует дополнительных ресурсов памяти и процессорного времени для поддержки его в актуальном состоянии. Однако, точных оценок трудно дать, так как многое зависит от конфигурации сервера, настройки ОС и СУБД и т.п.

Список литературы

- [1] Методы организации хранения данных в СУБД [Электронный ресурс]. – Режим доступа: <https://www.osp.ru/os/2003/03/182716>. – Дата доступа: 23.02.2022.
- [2] Дейт К. Дж. Введение в системы баз данных – Introduction to Database Systems. — 8-е изд. — М.: «Вильямс», 2006.
- [3] Josefsson S. The Base16, Base32, and Base64 Data Encodings. - The Internet Society, 2006
- [4] Организация хранения данных [Электронный ресурс]. – Режим доступа: https://studbooks.net/2267030/informatika/organizatsiya_hraneniya_dannyh. – Дата доступа: 23.02.2022.

UDC 004.921

FEATURES OF STORING DATA IN RELATIONAL DBMS

Kuzmin I.A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Piskun G.A. – PhD, associate professor, associate professor of the Department of ICSD

Annotation. The article discusses the basic principles of data storage when working with a DBMS. Some features of data storage in various DBMS are shown.

Keywords: *DB, Databases, DBMS.*