

# СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ IBM DB2

ЧУХАЕВ Артем

Технический Университет Молдовы

*Аннотация:* Статья посвящена анализу системы управления базами данных IBM DB2. Описано назначение данной системы, основные функциональные возможности, структура сервера баз данных, настройка конфигурации и организация хранения данных. Рассмотрены основные принципы работы сервера DB2, поддерживаемые виды объектов баз данных.

*Ключевые слова:* СУБД, IBM DB2, DB2 DPF, DGTТ, CGTT.

## 1. Возможности IBM DB2

**IBM DB2** — семейство систем управления реляционными базами данных, выпускаемых корпорацией IBM. Чаще всего, ссылаясь на DB2, имеют в виду реляционную систему управления базами данных DB2 UniversalDatabase. Язык SQL DB2 является вычислительно полным, то есть потенциально позволяет в декларативной форме определять любые вычисляемые соответствия между исходными данными и результатом. Это достигается, в том числе, за счёт использования табличных выражений, рекурсии и других развитых механизмов манипулирования данными.

IBM DB2 использует общепринятую на текущий момент клиент-серверную архитектуру реляционной СУБД, с обеспечением хранения информации на сервере и подключением приложений-клиентов к базам данных локально либо через сеть. В DB2 реализована поддержка всех распространенных промышленных стандартов доступа приложений к данным, включая стандартный язык запросов SQL, интерфейсы ODBC и JDBC, работу с типовыми текстовыми табличными форматами и т.п. Кроме того, DB2 включает в себя развитые возможности по хранению и работе с полуструктурированными данными в форматах XML, JSON/BSON.

Для разработки хранимых процедур в DB2 реализована поддержка множества процедурных языков, включая:

- стандартный для DB2 язык SQL,
- используемый в СУБД корпорации Oracle язык SQL/PL,
- возможность разработки «внешних» хранимых процедур на языках Java, C, C++ и COBOL.

Отличительными особенностями DB2 является:

- масштабируемость, ограниченная только доступными вычислительными ресурсами
- мощные встроенные средства разграничения и контроля доступа
- развитая интегрированная система резервного копирования и восстановления данных
- наличие полного набора технологий для построения «классических» аналитических хранилищ данных (деление таблиц на разделы, материализованные представления, оптимизации кэширования данных и сканирования таблиц и индексов, «внутренний» параллелизм исполнения сложных запросов и т.п.)
- поддержка построения конфигураций массивно-параллельной аналитической обработки данных (MPP) из множества серверов, соединенных через коммуникационную сеть, на базе DB2 DatabasePartitioningFeature (DB2 DPF)
- максимальная устойчивость к отказам и практически линейное масштабирование кластерных конфигураций DB2 pureScale, с хранением данных на общих дисках.

## 2. Как организовано хранение данных

Информация объектов базы данных (например, записи таблиц и элементы индексов) размещаются на страницах. Выделение дополнительного пространства для хранения данных выделяется группами страниц, которые называются экстентами. Выполнение операций выделения дополнительного пространства на уровне экстентов увеличивает производительность операций вставки и обновления записей.

Хранение информации в базах данных DB2 организовано в рамках объектов, называемых табличными пространствами. Табличное пространство представляет собой именованный набор контейнеров для хранения информации, размещаемых в файловой системе сервера баз данных.

Существуют следующие виды табличных пространств:

- обычные: используются для размещения пользовательских таблиц и индексов;
- большие: используются для размещения пользовательских таблиц и индексов, а также данных больших объектов (LOB) и данных XML. В современных версиях DB2 большие табличные пространства используются по умолчанию вместо обычных;
- временные: используются для размещения временной информации при выполнении запросов (системные временные табличные пространства) и временных таблиц, определенных приложениями (пользовательские временные табличные пространства).

Вид табличного пространства указывается при его создании и не может быть изменён иначе, как удалением и повторным созданием табличного пространства.

Начиная с DB2 версии 10.1, с целью упрощения управления хранением данных введено новое понятие – группа хранения данных (storagegroup). Группой хранения данных называется именованная совокупность путей в файловой системе сервера СУБД, которые могут использоваться для размещения данных. Состав групп хранения в базе данных обычно определяет набор видов устройств хранения, доступных для размещения информации. При создании базы данных в ней всегда автоматически создается группа хранения по умолчанию.

Каждое автоматически управляемое табличное пространство ассоциируется с одной из созданных групп хранения данных, что и определяет физическое размещение данных, хранящихся в соответствующих табличных пространствах. Существует возможность переместить табличное пространство из одной группы хранения в другую с помощью команды ALTER TABLESPACE ... USING STOGROUP ....

### 3. Виды объектов баз данных

**Схемы** – это пространства имён для сбора объектов базы данных. Схемы преимущественно используются для:

- обеспечения индикации монопольного использования объектов или связей с приложением;
- логической группировки связанных объектов.

Создание схемы может быть выполнено явно, путем вызова оператора CREATE SCHEMA, либо неявно, при первой попытке создания объекта без указания имени схемы. В последнем случае для успешного создания схемы пользователю должна быть предоставлена полномочие IMPLICIT\_SCHEMA.

**Таблица** – это собрание связанных данных, логически упорядоченных в столбцах и строках. Каждая строка таблицы состоит из одинакового набора именованных колонок. Каждой колонке при создании таблицы присваивается тип данных, ограничивающий допустимые значения колонки в строках таблицы (записях базы данных) и определяющий семантику возможных операций над соответствующими значениями (включая сравнение, сортировку, вычислительные операции).

Создание таблицы осуществляется командой CREATE TABLE, удаление – командой DROP TABLE. Поддерживается изменение описания таблицы командой ALTER TABLE, включая добавление и удаление колонок, изменение типов данных колонок. После выполнения некоторых операций изменения описания таблицы требуется выполнить её реорганизацию (реструктурировать физическое хранение таблицы для оптимального доступа к ней) с помощью команды REORG.

При создании таблицы существует возможность указать для колонок правила автоматического заполнения их значений. Особым случаем колонок с автоматическим заполнением являются столбцы идентификации – числовые столбцы, автоматически генерирующие уникальное числовое значение для каждой вставляемой строки. Автоматическое заполнение может осуществляться в одном из двух режимов:

- GENERATED ALWAYS – значение всегда устанавливается сервером DB2 и не может быть явно установлено приложением;
- GENERATED BY DEFAULT – значение устанавливается сервером DB2 в том случае, если приложение не указало явно присваиваемое значение при вставке записи.

Также на уровне таблицы могут быть определены ограничения, задающие ограничения на состав значений атрибутов. Поддерживаются следующие виды ограничений:

- первичный ключ (PRIMARY KEY) – ограничение уникальности на набор колонок, преимущественно используемых для поиска единичной записи, в таблице может быть только один первичный ключ;

- ограничение уникальности (UNIQUE) – дополнительное ограничение уникальности на набор колонок;
- внешний ключ (FOREIGN KEY) – ссылка в виде набора значений колонок, указывающая на комбинацию колонок другой таблицы, для которой определён внешний ключ или ограничение уникальности;
- проверка (CHECK) – логическое условие, ограничивающее возможные значения одной или сразу нескольких колонок в записи.

**Временные таблицы** - для хранения временных данных приложений в DB2 предусмотрен механизм временных таблиц, предоставляющий полный набор функций работы с табличными данными, но в контексте текущей пользовательской сессии.

Доступ к временным таблицам значительно повышает производительность, поскольку минимизируются либо вообще не возникают конфликты доступа к системному каталогу, а также не используется блокировка строк, ведение журнала (не обязательно, зависит от режима создания таблицы) и проверка полномочий.

Временные таблицы также поддерживают индексы, т. е. ко временной таблице можно создать любой стандартный индекс. Для таких таблиц можно также выполнить сбор статистик (командой RUNSTATS) для получения информации, необходимой оптимизатору запросов.

В DB2 существует две основные разновидности временных таблиц:

- объявленные временные таблицы (DGTT – Declared Global Temporary Tables);
- созданные глобальные временные таблицы (CGTT – CreatedGlobalTemporaryTables).

Объявленные временные таблицы — это созданные в памяти таблицы, используемые приложением и автоматически отбрасываемые при завершении его работы. Доступ к таким таблицам может получить только то приложение, которое их создало, и они не хранятся ни в одной из таблиц системного каталога DB2.

Каждая объявленная временная таблица имеет схему SESSION; эту схему необходимо указывать, ссылаясь на неё. Используемый для объявления временных таблиц идентификатор пользователя будет иметь в этих таблицах все привилегии. Каждое приложение, объявившее временную таблицу, будет иметь собственную копию такой таблицы.

**Триггер** — это объект, который автоматически выполняет операцию с таблицей или представлением. Определенное действие с объектом, для которого определен триггер, вызывает запуск триггера. Обычно триггер не считается объектом приложения, соответственно, обычно триггеры создаются не разработчиками, а администраторами баз данных.

**Хранимые процедуры** – это объекты базы данных, содержащие операторы SQL. Хранение части логики приложения в базе данных повышает производительность, поскольку сокращается объем трафика между приложением и базой данных. Кроме того, хранимые процедуры предоставляют централизованное местоположение для хранения программного кода, и, соответственно, другие приложения могут воспользоваться теми же хранимыми процедурами. Для вызова хранимой процедуры используется оператор CALL.

#### 4. Заключение

Круг задач, которые можно решать, используя объектные расширения DB2, существенно увеличен по сравнению с классическим реляционным подходом. Заимствовав из объектно-ориентированной модели наиболее необходимые для современных приложений элементы, DB2 сохранила все достоинства реляционной СУБД. Состояние СУБД должно регулярно отслеживаться и поддерживаться во избежание непредвиденных технических сбоев. Грамотно настроенная и оптимизированная СУБД гарантирует её высокую производительность, отказоустойчивость и безопасность критичных данных.

#### Литература

1. Техническая библиотека. [Электронный ресурс]. - Режим доступа: <https://www.ibm.com/developerworks/ru/library/>
2. Обзор возможностей IBMDB2. [Электронный ресурс]. - Режим доступа: <http://bourabai.kz/dbt/servers/DB2.htm>
3. IBM DB2. [Электронный ресурс]. - Режим доступа: [https://en.wikipedia.org/wiki/IBM\\_Db2](https://en.wikipedia.org/wiki/IBM_Db2)
4. DB2 Database. [Электронный ресурс]. - Режим доступа: <https://www.webopedia.com/TERM/D/DB2.html>
5. IBM DB2 Introduction. [Электронный ресурс]. - Режим доступа: <http://www.cs.umd.edu/class/spring2002/cmsc434-0101/MUIseum/applications/db2.html>