



# ОРГАНИЗАЦИЯ SQL-СЕРВЕРА

***МДК 02.01 Информационные  
технологии и платформы  
разработки информационных  
систем***

# Сервер баз данных



**Сервер баз данных** - программное образование, привязанное к соответствующей базе (базам) данных, существующее независимо от существования пользовательских (клиентских) процессов и выполняемое, (не обязательно) на выделенной аппаратуре.

# Базовая архитектура сервера баз данных

Доступ к базе данных от прикладной программы или пользователя производится путем обращения к клиентской части системы. В качестве основного интерфейса между клиентской и серверной частями выступает язык баз данных SQL. *Собирательное название SQL-сервер относится ко всем серверам баз данных, основанных на SQL.*

## Функции SQL-сервера:

- поддержание логически согласованного набора файлов;
- обеспечение языка манипулирования данными;
- восстановление информации после сбоев;
- организация параллельной работы нескольких пользователей



Сервер базы данных

# Типы серверов баз данных

- **SQL-серверы** (SQL-процессоры: MS SQL Server, Oracle Server, Sybase SQL Server, Informix Online );
- **объектно-ориентированные** (*GemStone* компании *GemStone Systems*, хранят информацию в форме объектов, что больше подходит для средств разработки программ-клиентов, которые работают с чисто объектной моделью и в них лучше, чем в реляционных, организовано хранение сложных структур данных);
- **многомерные** (*Essbase Analysis Server* фирмы *Arbor Software Corp.*, обеспечивают более понятное конечным пользователям представление данных и обладают новыми средствами оперативного анализа OLAP).

# Базовая архитектура сервера баз данных. Физическая структура.

Любой SQL-сервер имеет физическую и логическую структуру.

**Физическая структура** - это метод, с помощью которого SQL-сервер хранит свои данные в ОС.

На настоящий момент известно 2 варианта хранения данных:

- **с буферизацией средствами ОС** (сервер СУБД держит свои данные в файлах);
- **без буферизации** (сервер СУБД работает со своими данными на отдельном разделе диска - raw access, и доступ к которому полностью управляется сервером СУБД).

# Использование серверов БД

1. **Системы с онлайн-обработкой транзакций (OLTP - Online Transaction Processing** - способ организации работы СУБД при котором система работает с транзакциями небольшими по размерам, но идущими большим потоком, и при этом клиенту требуется от системы максимально быстрое время ответа).
2. **Системы поддержки принятия решений (DSS - Decision Support System** - характеризуются тем, что в них преобладают массивные выборки по большому объему данных в режиме "только чтение").
3. **Системы пакетной обработки (BPS - Batch processing System** - есть некоторая очередь заданий, каждое из заданий выполняется с полным доступом к ресурсам и максимальной эффективностью.
4. Использование серверов СУБД в качестве **медиа серверов (MMS - Multi Media Servers** - сервер используется для предоставления пользователям доступа к аудио и видео информации. Задача сервера - обеспечение непрерывности мультимедиа потока во избежание рывков видео и прерывания звука вплоть до уровня передачи в канал).

# Логическая организация сервера

**Логическая структура сервера** - это те объекты СУБД, которыми сервер СУБД непосредственно манипулирует (но не в контексте операционной системы, а контексте базы данных и которые "видны" пользователю).

1. **Таблица** - это основная единица хранения информации в системе (пользовательские таблицы и системные таблицы - data dictionary -словарь данных). С таблицами связаны правила целостности данных (***data integrity rules***), которые определяются ограничителями (***constraint***) и триггерами (***trigger***).
2. **Представления** (view - сгруппированные данные из нескольких таблиц; в представления можно вставлять строки данных, удалять их оттуда и модифицировать, с теми ограничениями, которые действуют для базовых таблиц или для реализации сервера СУБД ).

# Логическая организация сервера

- 3. Программные модули** (сохраненная процедура - некоторый откомпилированный код, который лежит непосредственно на сервере СУБД и используется по мере надобности прикладными программами).
- 4. Синонимы** (алиасы для таблицы, представления или программного модуля - ссылка на объект, используется для того, чтобы скрыть реальное имя объекта или реального пользователя объекта, обеспечения общего доступа к объекту или прозрачности доступа к объектам удаленной базы данных).
- 5. Индексы** (создаются для ускоренного поиска по таблице, сервер целиком управляет индексом при операциях с таблицей и нет необходимости специально перестраивать или корректировать индексы, при обработке запроса SQL-сервер сам может выбирать подходящий индекс из доступных для данной таблицы.).

# Логическая организация сервера

6. **Кластер** (*cluster* – связка таблиц в кластеры применяется для сокращения времени выполнения запросов, использование кластеров зависит от реализации SQL-сервера).
7. **Последовательность** (*sequence* - объект СУБД специфичен для Oracle и представляет собой специальный объект БД для генерации последовательных значений. Каждое следующее обращение к последовательности увеличивает ее текущее значение на шаг, определяемый при создании последовательности. Использование последовательностей зависит от реализации SQL-сервера).

# Поддержка целостности данных

Главная особенность SQL-технологий - наличие у сервера БД *специальных средств контроля целостности данных*, не зависящих от клиентских программ и привязанных непосредственно к таблицам.

К структурам контроля целостности данных относятся:

- *ограничители (constraint)*, которые привязаны к столбцам;
- *триггеры (trigger)*, которые могут быть привязаны как к столбцам, так и к строкам в таблице.

# Поддержка целостности данных

**Ограничители** - это элементарные проверки или условия, которые выполняются для операций вставки и модификации значения столбца.

Если данная проверка не проходит или условие не выполняется, то вставка или модификация отменяется, а в программу клиента передается ошибка.

SQL-серверы, как правило, поддерживают следующие ограничители:

- **NOT NULL** - проверка на непустое значение.
- **UNIQUE** - проверка на уникальность.
- **PRIMARY KEY** - первичный ключ.
- **ограничение ссылочной целостности (referential integrity constraint)** не позволяет значениям из столбца одной таблицы принимать значения кроме как из присутствующих в столбце другой таблицы. Это делается при помощи ограничителей **FOREIGN KEY** (внешний ключ) и **REFERENCES** (указатель ссылки).
- **CHECK** - проверка фиксированного условия.

# Поддержка целостности данных

**Триггеры** (*triggers*) - это сохраненная откомпилированная процедура, которая связана с определенной таблицей.

Триггеры, в отличие от ограничителей, могут выполнять сколь угодно сложные манипуляции над данными. Можно также задавать порядок срабатывания триггера относительно операции, т.е. выполниться ли триггер перед операцией вставки/модификации/удаления значения из столбца (или всей строки) или непосредственно после такой операции.

Применение триггеров:

- **Прозрачный аудит** (не зависящий от клиентских программ и невидимый для них) и регистрация событий, связанных с доступом к определенным таблицам или столбцам в таблицах.
- **Генерация значений в столбцах** на основе значений в других столбцах при вставке/модификации строки данных.
- **Манипуляции над зависимыми таблицами** в особенности, если они находятся на других узлах распределенной базы данных, чего нельзя сделать при помощи ограничителей.

В случае необходимости триггеры можно запрещать, а затем разрешать. Запрещение триггеров применяется обычно при массовых загрузках данных в таблицы извне, с целью уменьшения времени загрузки.

# Способы управления данными

1. **Атомарность SQL-выражений** при работе с данными - неизменность (фиксация во времени) набора данных, с которыми это выражение работает на всем протяжении своего исполнения: при выполнении оператора языка SQL над определенной таблицей состояние таблицы на момент начала выполнения операции фиксируется во времени и не изменяется до конца выполнения оператора.

Этот набор данных для текущего выполняемого выражения не может быть изменен другим пользователем или даже другой сессией этого же пользователя, которая пытается выполнить операцию модификации этих же данных в этой же таблице.

2. **Распараллеливание операций:** типовые операции с таблицей в базе данных состоят из многих однотипных операций, каждая из которых может быть выполнена независимо.

В связи с этим такие операторы очень хорошо распараллеливаются при использовании многопроцессорных систем, что позволяет выровнять нагрузку в системе между разными процессорами, при том условии, что сервер БД умеет работать в многопроцессорной конфигурации, и уменьшить время ответа системы.

# Способы управления данными

3. *Обеспечение максимальной производительности* достигается за счет реализации следующих принципов:
- процессы, выполняющие чтение блоков данных, никогда не ожидают процессов, выполняющих запись тех же блоков данных.
  - процессы, выполняющие запись блоков данных, при отсутствии явных блокировок со стороны пользователя, не ожидают процессов, выполняющих чтение тех же блоков данных.
  - процессы, выполняющие запись блоков данных, ожидают другие процессы, выполняющие запись, только в случае если они пытаются выполнить запись данных в одни и те же блоки данных.

## 4. *Транзакции.*

Транзакция - это логическая единица работы, которая состоит из одного или нескольких SQL-выражений. Группа выражений отмеченных как транзакция рассматривается как единое и неделимое целое. В случае если в одном из выражений обработки данных происходит ошибка, то транзакция отменяется целиком.

# Способы управления данными

## 5. *Блокировки.*

Блокировки по аналогии с базами данных на основе файлов могут быть как разделяемые, так и исключительные. Блокировки связаны с транзакциями. Если выполняется отмена транзакции, то снимаются все связанные с этой транзакцией блокировки.

Многие блокировки выполняются неявно для пользователя, они выставляются, например, операторами UPDATE, INSERT. Существуют явные операторы задания блокировок, например, LOCK TABLE.

*Многие SQL-серверы имеют специальные способы обнаружения и предотвращения взаимных блокировок (deadlocks), которые могут занимать ресурсы СУБД на неопределенное время.*

*Способы, которыми обеспечиваются блокировки, зависят от реализации сервера, и описываются в его документации. Виды блокировок также зависят от используемого сервера.*

# Способы управления данными

## 6. *Политика безопасности.*

Эта задача решается при помощи идентификации пользователей внутри сервера БД, раздачи этим пользователям соответствующих прав доступа на объекты внутри БД, а также при помощи аудита доступа к объектам БД.

7. *Лицензирование* – защита программных продуктов при помощи серийного номера и ключа активации, при этом количество одновременно активных пользователей БД может быть зашито в серийный номер.