

Современные
технологии баз
данных и анализа
информации- 1

Функциональность и архитектура СБД

Понятие серверных БД

- Серверные СУБД (сервер БД-СБД) - СУБД, реализующие технологию «клиент-сервера» (CS – Client-Server, CS DBMS)
 - Примеры СБД:
 - СУБД Oracle;
 - СУБД Microsoft SQL Server;
 - СУБД MySQL (Oracle);
 - PostgreSQL;
 - IBM DB2 и др.
 - Каждый СБД может работать на определенных типах компьютеров и сетей. ОС серверов могут быть Windows Server, Unix, Linux и др.
-



Функциональность СБД

- обслуживание БД;
- управление БД;
- обеспечение целостности и сохранности данных;
- обеспечение операций ввода-вывода при доступе клиента к информации.



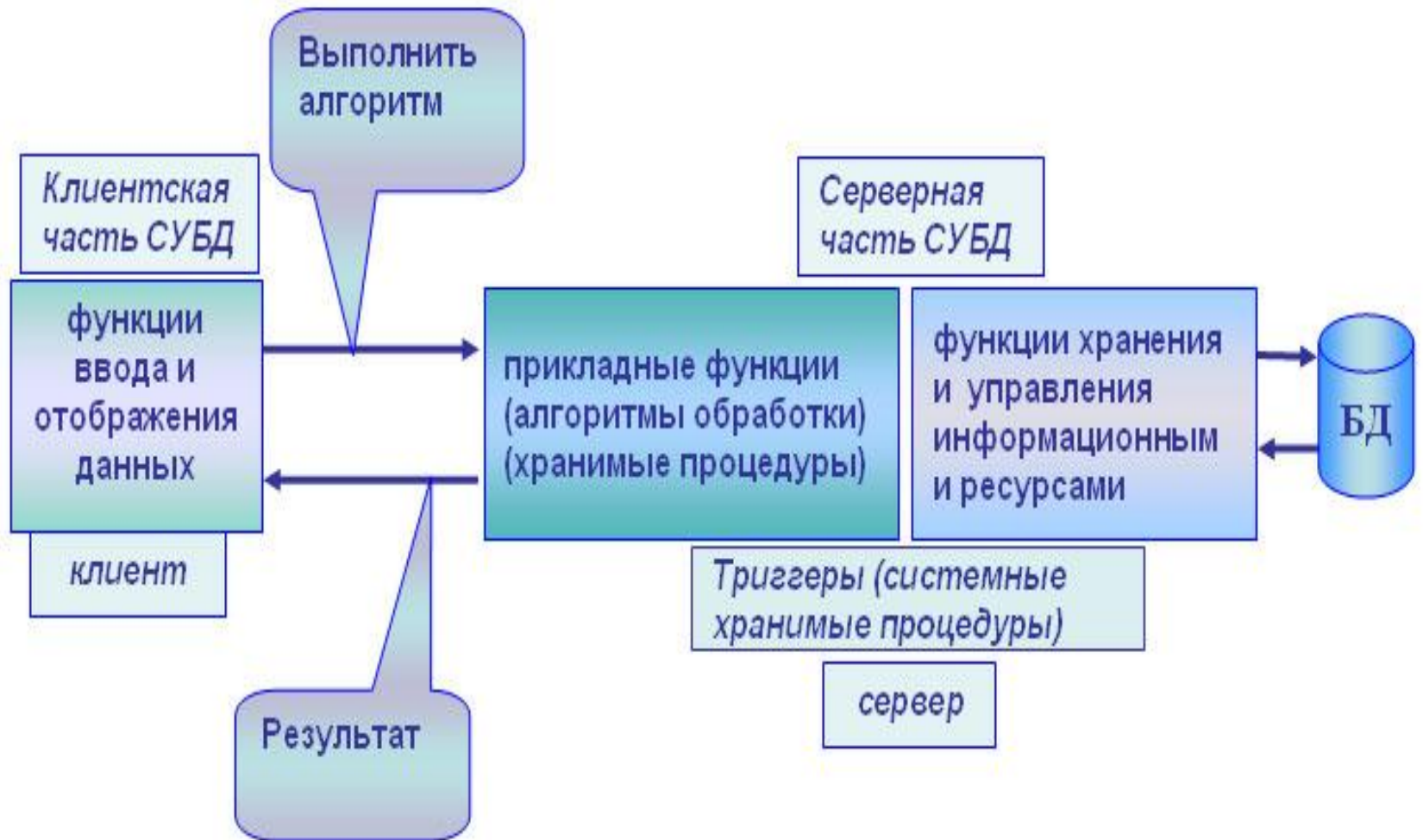
Архитектура СБД

Состоит из следующих уровней:

- Внутренний уровень архитектуры - наиболее близок к физическому уровню СУБД, описывает способ размещения данных на устройствах хранения информации. Для пользователя к просмотру и модификации не доступен
- Внешний уровень архитектуры - наиболее близок к пользователю, предоставляет возможность манипуляции данными в СУБД с помощью языка SQL.



Модель типового СБД



Физическая структура типового СБД

Подсистема взаимодействия с клиентским приложением

Подсистема компиляции запросов

Подсистема планирования и выполнения запросов

Подсистема выполнения транзакций

Подсистема управления памятью

Подсистема взаимодействия с клиентским приложением

Данный модуль отвечает за поддержание связи с клиентом. Как правило, механизм его работы выглядит следующим образом. Подсистема взаимодействия "прослушивает" сеть в ожидании клиентских запросов на установление соединения. Когда такой запрос обнаруживается, порождается новый процесс, который будет обеспечивать связь с данным клиентом. Клиенту сообщается идентификатор данного процесса, в дальнейшем клиент передает свои запросы и получает данные взаимодействуя с этим интерфейсным процессом. После того, как клиент закрывает соединение, обслуживавший его процесс прекращается. Характеристики интерфейсных процессов зависят от операционной системы, в которой работает СБД.



Подсистема компиляции запросов

Данный модуль отвечает за компиляцию поступающих от клиентов через интерфейсные процессы запросов во внутренний код, который будет исполняться сервером. При ошибках компиляции соответствующие сообщения передаются клиенту. Наиболее современные СУБД позволяют сохранять откомпилированный код запросов некоторое время. Это позволяет избежать стадии компиляции при повторном обращении клиента к запросу.



Подсистема планирования и выполнения запросов

Данный модуль должен составить такой план выполнения запроса, чтобы он был обработан наиболее быстро. Для этого анализируются условия выборок и соединений, устанавливается порядок их выполнения.

Планировщики запросов ведущих СУБД отслеживают информацию о распределении значений в таблицах. План выполнения запроса включается в его откомпилированный код.



Подсистема выполнения транзакций

Здесь выполняется оптимизированный код запроса, обновляются индексы, выполняются в случае необходимости триггеры и хранимые процедуры. Как правило, несколько запросов могут исполняться параллельно, при этом обеспечивается необходимый уровень их изоляции. Также ведется журнал транзакций, обеспечивается их завершение и корректный откат.



Подсистема управления памятью

Этот компонент отвечает за считывание данных с диска в оперативную память, синхронизацию обновлений с данными на диске и т.д. Он может использовать файловые функции операционной системы, но часто СУБД имеет свои собственные низкоуровневые средства доступа к дискам.



Логически данные в СБД организованы в виде объектов

Tables	Таблицы базы данных, в которых хранятся собственно данные
Views	Представления для отображения данных из таблиц
Stored Procedures	Хранимые процедуры
Triggers	Триггеры – специальные хранимые процедуры, вызываемые при изменении данных в таблице
User Defined function	Создаваемые пользователем функции
Indexes	Индексы – дополнительные структуры, призванные повысить производительность работы с данными
User Defined Data Types	Определяемые пользователем типы данных
Keys	Ключи – один из видов ограничений целостности данных
Constraints	Ограничение целостности – объекты для обеспечения логической целостности данных
Users	Пользователи, обладающие доступом к базе данных
Roles	Роли, позволяющие объединять пользователей в группы
Rules	Правила базы данных, позволяющие контролировать логическую целостность данных
Defaults	Умолчания или стандартные установки базы данных

Объекты СБД

Таблицы (Tables)

- Все данные в СБД содержатся в таблицах. Таблицы представляют собой совокупность каких-либо сведений об объектах, явлениях, процессах реального мира. Никакие другие объекты не хранят данные, но они могут обращаться к данным в таблице. Таблицы содержат:
 - строки; каждая строка (или запись) представляет собой совокупность атрибутов (свойств) конкретного экземпляра объекта;
 - столбцы; каждый столбец (поле) представляет собой атрибут или совокупность атрибутов. Поле строки является минимальным элементом таблицы. Каждый столбец в таблице имеет определенное имя, тип данных и размер.

Представления (View)

- ▣ **Виртуальная таблица**, представляющая собой поименованный запрос
- ▣ **Динамический результат** одной или нескольких реляционных операций над базовыми отношениями с целью создания некоторого иного отношения.



Преимущества представлений

- **Независимость от данных** - с помощью представлений можно создать согласованную, неизменяемую картину структуры базы данных, которая будет оставаться стабильной даже в случае изменения формата исходных таблиц
 - **Актуальность** - изменения данных в любой из таблиц базы данных, указанных в определяющем запросе, немедленно отображаются на содержимом представления
 - **Снижение стоимости** - представления позволяют упростить структуру запросов за счет объединения данных из нескольких таблиц в единственную виртуальную таблицу. В результате многотабличные запросы сводятся к простым запросам к одному представлению
 - **Повышение защищенности данных** - права доступа к данным могут быть предоставлены исключительно через ограниченный набор *представлений*, содержащих только то подмножество данных, которое необходимо пользователю. Подобный подход позволяет существенно ужесточить контроль за доступом отдельных категорий пользователей к информации в базе данных
-
- 

Курсоры (Cursor)

- **Курсор** - указатель на определенную строку в результирующем наборе данных
- **Курсор** - это область в памяти базы данных, которая предназначена для хранения последнего оператора SQL



Хранимые процедуры (**Stored procedure**)

- ▣ **Хранимая процедура** – объект базы данных, представляющий собой набор SQL-инструкций, который компилируется один раз и хранится на сервере



Хранимые процедуры

- обрабатывают входные параметры и возвращают вызывающей программе значения в виде **выходных** параметров;
- содержат программные инструкции, которые выполняют операции в базе данных, включая вызов других процедур;
- возвращают значение состояния вызывающей программе, таким образом передавая сведения об успешном или неуспешном завершении (и причины последнего).



Особенности

- Хранимая процедура относится не к отдельному объекту, а ко **всей БД в целом**
- Хранимые процедуры **поддерживают модульное программирование**, так как позволяют разбивать большие задачи на самостоятельные, более мелкие и удобные в управлении части



Преимущества ХП

- Снижение сетевого трафика между клиентами и сервером
- Большая безопасность
- Повторное использование кода
- Более легкое обслуживание
- Повышенная производительность



Триггеры (Trigger)

- Триггер — это (составной) оператор SQL, который автоматически выполняется в СУБД, когда происходит некоторое заданное событие, называемое триггерным событием (triggering event).



Основные виды триггеров

- DML триггер - привязан к конкретной таблице или представлению и определяется для конкретного события (INSERT, UPDATE и DELETE)
- DDL триггер - это особый вид триггеров, которые срабатывают при выполнении инструкций языка описания данных DDL (CREATE, ALTER, DROP, GRANT, DENY, REVOKE). Они могут применяться при выполнении административных задач (например, для аудита и регулирования операций в базе данных)



Использование триггеров

- проверка правильности введенных данных и проверка выполнения сложных ограничений целостности данных, которые трудно, если вообще возможно, поддерживать с помощью ограничений целостности
- выдача предупреждений (например, с помощью электронной почты), которые напоминают о необходимости выполнить некоторые действия при обновлении таблицы
- накопление информации аудита посредством фиксации сведений о внесенных изменениях и тех лицах, которые их выполнили
- поддержка репликации



Недостатки триггеров

- при перемещении некоторых функций в БД усложняются задачи ее проектирования, реализации и администрирования



Различия между триггерами и ХП

- триггеры нельзя вызывать из кода программы. СБД вызывает их автоматически в ответ на определенное событие
- триггеры не имеют списка параметров
- спецификация триггера немного отличается от спецификации процедуры
- триггеры не привязаны ко всей БД

