

## Тема 28. Распределенная обработка данных.

### *Модели «клиент-сервер»*

В сетевой среде пользователи и базы данных могут находиться на разных компьютерах, при этом сама база может быть централизованной (размещаться на одном компьютере) или децентрализованной (занимать несколько компьютеров). Параллельный доступ нескольких пользователей к централизованной базе данных называется распределенной обработкой данных. Параллельный доступ нескольких пользователей к децентрализованной БД – распределенная база данных.

Модель «клиент-сервер» разделяет процесс решения программной задачи на два взаимодействующих процесса: клиента и сервера. Клиент запрашивает некоторые услуги у сервера, а сервер обеспечивает их выполнение. Предполагается, что серверный процесс одновременно обслуживает множество клиентских процессов.

Стандартное интерактивное приложение в общем случае выполняет функции следующих видов:

1. функции ввода и отображения данных (представительские функции);
2. функции обработки данных;
3. функции управления данными.

Представительские функции реализуют графический интерфейс пользователя.

Функции обработки данных определяют алгоритмы решения прикладных задач.

Управление данными – хранение и управление базами данных (СУБД).

Возможные распределения функций интерактивного приложения между процессами клиент и сервер:

	Представление	Обработка данных	Управление данными
Распределенное представление	Клиент Сервер	сервер	сервер

Удаленное представление	Клиент	сервер	сервер
Распределенная обработка	Клиент	клиент сервер	сервер
Удаленное управление данными	Клиент	клиент	сервер
Распределенное управление данными	Клиент	клиент	клиент сервер

В модели распределенного представления основная часть работы выполняется на сервере, клиент работает в режиме терминала и реализует только часть представительских функций.

Модель удаленного представления принято называть моделью сервера баз данных.

Распределенная обработка характеризуется выполнением прикладных алгоритмов и на клиенте, и на сервере.

Удаленное управление принято называть удаленным доступом к данным.

Распределенное управление данными - антипод распределенного представления: большая часть обработки происходит на клиенте.

### Модель удаленного доступа к данным

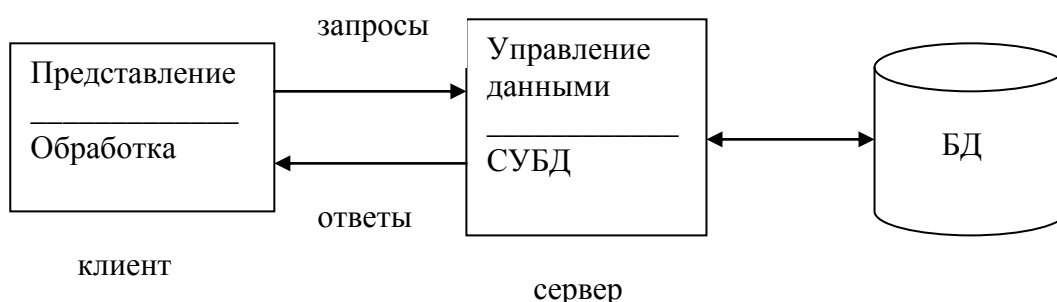


Рис. 28.1. Модель удаленного доступа к данным.

База данных хранится на сервере, там же находится ядро СУБД. На клиенте реализуются функции представления и обработки. Клиент обращается к серверу с запросами на SQL.

Достоинством модели является большое число СУБД, поддерживающих удаленный доступ к данным. Недостатки: большая загрузка сети, дублирование

функций обработки разных приложений, сложность разработки приложений, пассивность сервера.

### Модель сервера баз данных

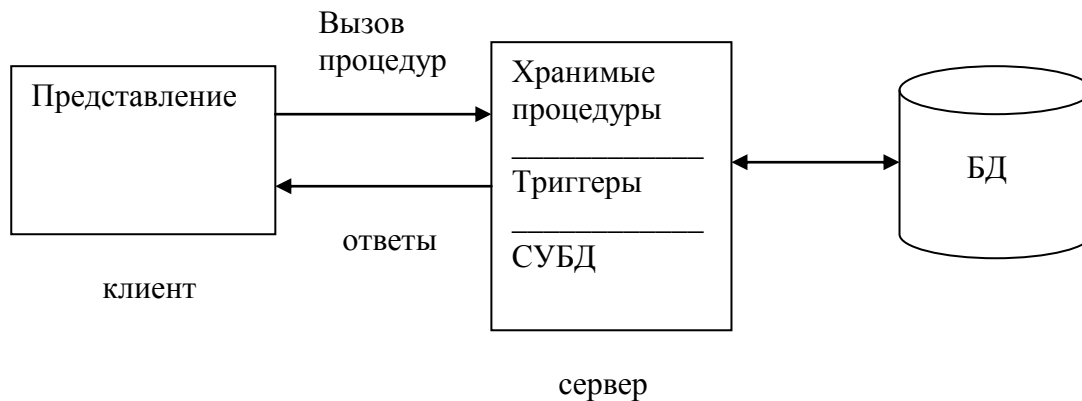


Рис. 28.2. Модель сервера БД.

Основу модели составляет механизм хранимых процедур и триггеров (рис. 28.2). Обработка данных разделена между клиентом и сервером. Со стороны сервера обработка поддерживается хранимыми процедурами, которые хранятся в базе данных и управляются СУБД. Клиентское приложение запускает хранимую процедуру и получает запрошенные данные.

Контроль состояния БД выполняется с использованием триггеров – процедур, автоматически запускаемых при возникновении некоторых состояний. Триггеры могут вызывать хранимые процедуры.

В этой модели сервер активен, он может быть инициатором обработки данных. Для написания хранимых процедур и триггеров используется SQL.

Недостаток модели – большая загрузка сервера.

### Модель сервера приложений

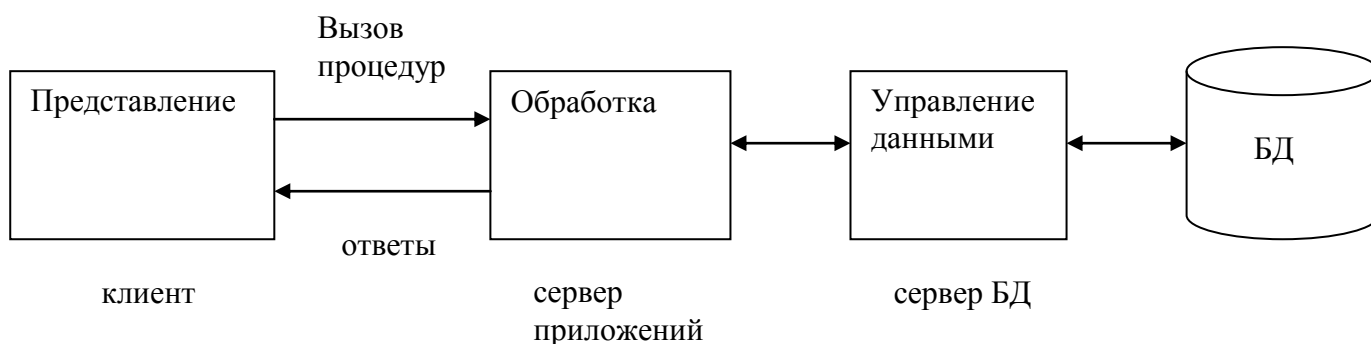


Рис. 28.3. Модель сервера приложений.

Это трехуровневая модель - функции приложения распределены между тремя процессами: клиентом, сервером приложений и сервером БД (рис. 28.3).

Клиент выполняет функции представления. Серверы приложений реализуют общие функции для клиентов, поддерживают распределенные транзакции. Серверы баз данных обеспечивают создание и ведение БД, поддерживают целостность данных.

Модель более гибкая, чем двухуровневая. Преимущества модели особенно проявляются в случае сложной обработки данных.

## Базы данных в Internet и intranet

### Intranet

Технология intranet – это технология Internet, перенесенная в среду локальных вычислительных систем (корпоративных информационных систем). Корпоративные системы intranet, в отличие от систем «клиент-сервер», ориентированы не на данные, а на информацию в ее окончательном и пригодном для использования неквалифицированным пользователем виде.

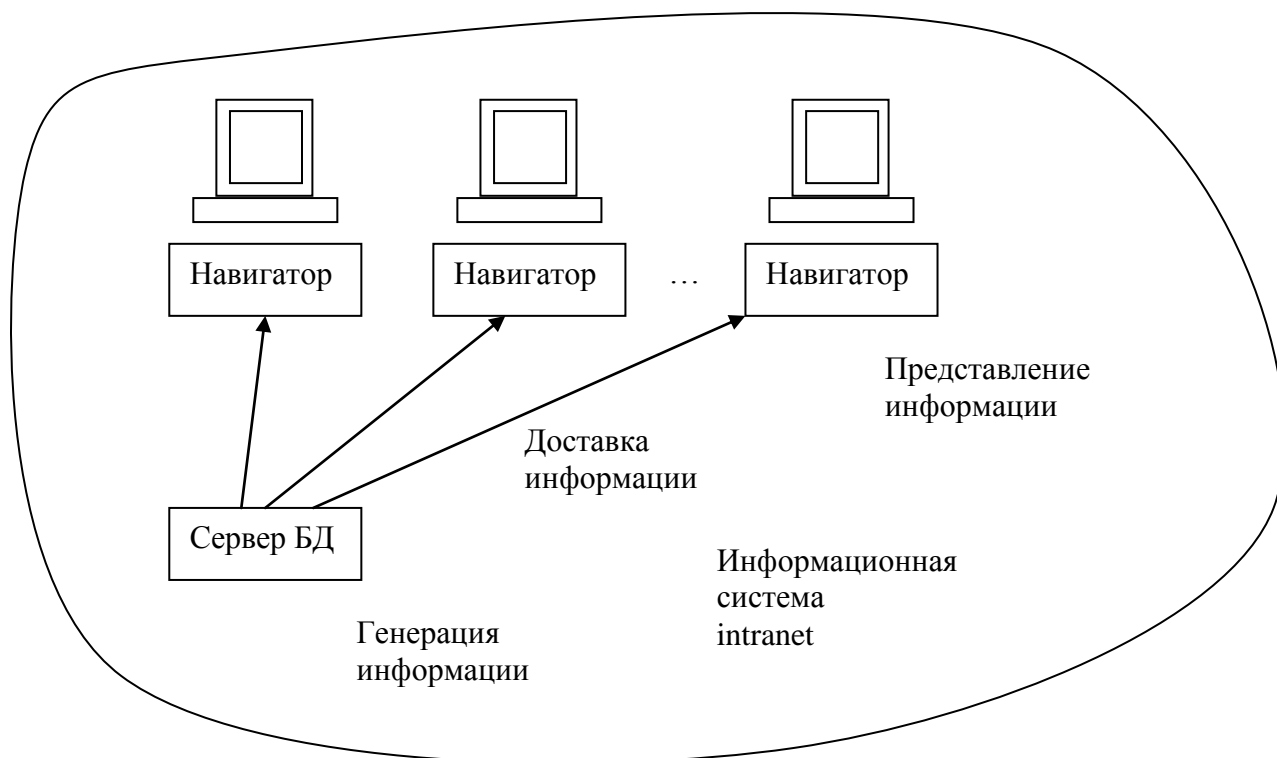


Рис. 28.4. Информационная система intranet.

Отличительные черты технологии intranet:

- на сервере порождается информация, а не данные;
- при обмене между клиентом и сервером используется протокол открытого стандарта;
- прикладная система находится на сервере, поэтому на компьютерном клиенте достаточно иметь навигатор.

### Доступ к базам данных в Internet

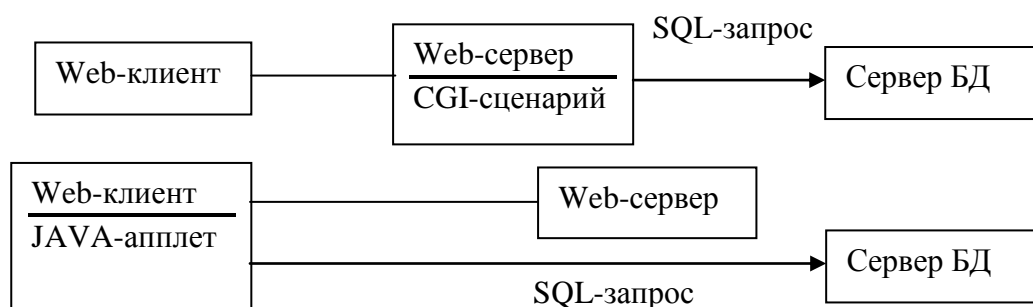


Рис. 28.5. Базы данных в Internet.

Программы WWW могут иметь доступ к БД как со стороны сервера, так и со стороны клиента (рис. 28.5).

При доступе со стороны сервера обращение к БД обычно производится путем вызова Web-сервером внешних программ в соответствии с соглашениями одного из интерфейсов: CGI (интерфейс общего шлюза), API (интерфейс прикладного программирования). Внешние программы взаимодействуют с сервером БД на SQL. Они пишутся на обычных языках программирования или на специализированных языках типа Perl или PHP. Такие программы называются CGI-сценариями или CGI-скриптами. Для поддержания этого механизма на стороне клиента в состав HTML-документа включается форма для представления запросов к БД.

Доступ в соответствии с интерфейсом CGI включает следующие шаги:

1. Запрос клиентом у сервера страницы, содержащей форму с обращением к БД.

2. Заполнение формы клиентом и отправка ее серверу.
3. Сервер, получив форму, запускает CGI-сценарий и передает ему параметры формы.
4. CGI-сценарий преобразует параметры запроса в SQL-запрос и направляет его серверу БД.
5. Получив результаты запроса, сценарий формирует HTML-страницу, передает ее серверу и завершается.
6. Сервер передает сформированную страницу клиенту.

Такой подход имеет как достоинства (языковая независимость, широкая распространенность и др.), так и недостатки (многократные установки и разрывы соединений с БД, большое время выполнения сценариев).

В технологии API внешние программы оформляются в виде динамических библиотек, что имеет и достоинства и недостатки. Достоинства: ускорение выполнения программы, большая функциональность. Недостатки: языковая зависимость, слабая защита сервера, непереносимость на другие платформы.

При доступе к БД со стороны клиента используются языки Java и JavaScript.

Последовательность действий для доступа к данным:

1. На языке Java пишутся программы-апплеты.
2. В тексте HTML-документа ставятся ссылки на соответствующие апплеты. Сами апплеты хранятся на сервере.
3. При обнаружении на странице ссылки происходит передача апплета с сервера на компьютер клиента и его загрузка.
4. Получив управление, апплет взаимодействует с сервером БД и предоставляет затребованную информацию пользователю.

Для обращения к серверу БД используется стандарт JDBC, основанный на ODBC.