Вариант 3

1. Укажите преимущества и недостатки локальных баз данных.
2. Назовите основные характеристики архитектуры «клиент—сервер».
3. Расскажите о двухзвенной модели архитектуры «клиент—сервер».
4. Дайте сравнительную характеристику трех- и двухзвенной моделей архитектур «клиент—сервер».
5. Назовите четыре состояния транзакции, приведите их особенности.
6. **Недостатки:**

* ограниченное число пользователей;
* длительное время ожидания ответа.

**Преимущества:**

* расположение на клиенте PL и BL функций существенно разгружает сервер БД, сводя к минимуму общее число процессов в ОС
* процессор сервера целиком загружается операциями обработки данных, запросов и транзакций
* резко уменьшается загрузка сети, т.е в ответ на запросы клиент получает только необходимые данные
* стандартным при обращении приложения клиента к серверу становится язык SQL.

1. Основной принцип технологии "клиент-сервер" заключается в разделении функций приложения на три группы:

* ввод и отображение данных (взаимодействие с пользователем);
* прикладные функции, характерные для данной предметной области;
* функции управления ресурсами (файловой системой, базой даных и т.д.)

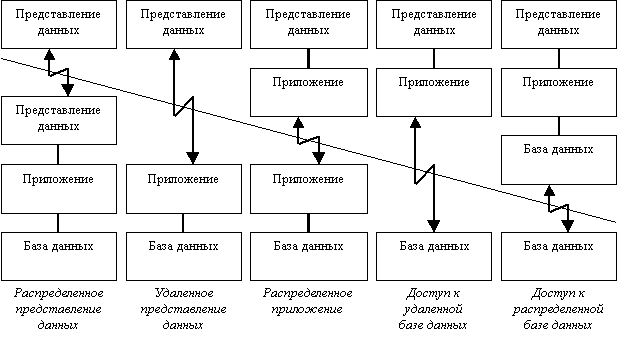
1. **Двухзвенная модель**(two-tier model) – это система "клиент-сервер", в которую входят компьютеры клиента и сервера. Клиент запрашивает данные у сервера, а сервер предоставляет данные. Большинство систем "клиент-сервер" построены с использованием этой модели, но двухзвенные модели способны обеспечить работу лишь ограниченного числа клиентов.

Двухзвенная модель "клиент-сервер" подходит для небольших программ на уровне рабочей группы при числе пользователей менее 100 (конечно, в зависимости от того, что делают прикладные программы). В большинстве двухзвенных систем невозможно существенно увеличить это число.

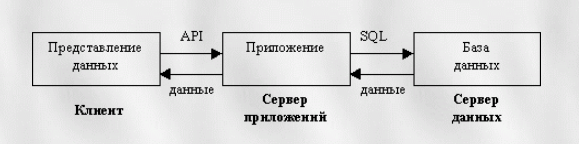
1. Сравнительная характеристика

|  |  |
| --- | --- |
| Двухзвенная модель | Трёхзвенная модель |
| **Преимущества:**   * расположение на клиенте PL и BL функций существенно разгружает сервер БД, сводя к минимуму общее число процессов в ОС * процессор сервера целиком загружается операциями обработки данных, запросов и транзакций * резко уменьшается загрузка сети, т.е в ответ на запросы клиент получает только необходимые данные * стандартным при обращении приложения клиента к серверу становится язык SQL. | *Преимущества:*   * уменьшился сетевой трафик; * уменьшилась сложность программного обеспечения приложения клиента; * понизились требования к аппаратным средствам клиента; * повысились надежность, целостность и секретность базы данных. |
| **Недостатки:**   * ограниченное число пользователей; * длительное время ожидания ответа. | *Недостатки:*  • очень большая стоимость сервера. |

Двухзвенная модель:



Трёхзвенная модель:



1. **4 состояния транзакции:**

**Активное**: Такое состояние имеет транзакция, которая

1. не стартовала.
2. стартовала, но еще не завершилась.
3. стартовала, но не закончилась вызовом любой процедуры завершения. (Например, из-за сбоя питания, обрыва соединения и т. п.)

**Отмененное**: Транзакция, которая

1. завершилась процедурой ROLLBACK, т. е. запросила удалить все произведенные ею изменения из БД.
2. была помечена как Активная, но была обнаружена в неживом состоянии другой транзакцией, которая и помечает ее как отмененную.

В любом случае, изменения, сделанные такой транзакцией, должны быть игнорированы и удалены из базы данных

**Подтвержденное**: Транзакция, которая завершила всю свою работу успешно

1. вызовом процедуры COMMIT
2. вызовом процедуры Rollback, но не произведя никаких изменений в БД.

В любом случае транзакция завершилась, и никогда не возобновится снова. Ее изменения теперь являются частью корректного состояния БД.

**Limbo**: Транзакция, стартовавшая в режиме 2PC (two phase commit) вызовом процедуры Prepare. Эта транзакция может быть живой или нет. В любой момент такая транзакция может возобновиться и запросить подтверждение или отмену. Изменения, произведенные транзакцией, оставшейся в состоянии in limbo не могут быть приняты или игнорированы, соответственно они не могут быть удалены из БД.