

1. Да. Клиентский процесс запрашивал некоторые услуги, а серверный процесс обеспечивал их выполнение. При этом предполагалось, что один серверный процесс может обслужить множество клиентских процессов.

2. Процессор управления данными (Database Manager System Processing) — это собственно СУБД, которая обеспечивает хранение и управление базами данных. В идеале функции СУБД должны быть скрыты от бизнес-логики приложения, однако для рассмотрения архитектуры приложения нам надо их выделить в отдельную часть приложения.

В централизованной архитектуре (Host-based processing) эти части приложения располагаются в единой среде и комбинируются внутри одной исполняемой программы.

В децентрализованной архитектуре эти задачи могут быть по-разному распределены между серверным и клиентским процессами.

3. Модель удаленного управления данными также называется моделью файлового сервера (File Server, FS). В этой модели презентационная логика и бизнес-логика располагаются на клиенте. На сервере располагаются файлы с данными и поддерживается доступ к файлам. Функции управления информационными ресурсами в этой модели находятся на клиенте. Достоинства этой модели в том, что мы уже имеем разделение монопольного приложения на два взаимодействующих процесса. При этом сервер (серверный процесс) может обслуживать множество клиентов, которые обращаются к нему с запросами. Собственно СУБД должна находиться в этой модели на клиенте.

4. В модели «файл-сервер» высокий сетевой трафик, который связан с передачей по сети множества блоков и файлов, необходимых приложению

5. Недостатки данной модели:

- Все-таки запросы на языке SQL при интенсивной работе клиентских приложений могут существенно загрузить сеть;
- Так как в этой модели на клиенте располагается и презентационная логика, и бизнес-логика приложения, то при повторении аналогичных функций в разных приложениях код соответствующей бизнес-логики должен быть повторен для каждого клиентского приложения — это вызывает излишнее дублирование кода приложений;
- Сервер в этой модели играет пассивную роль, поэтому функции управления информационными ресурсами должны выполняться на клиенте. Действительно, например, если нам необходимо выполнять контроль страховых запасов товаров на складе, то каждое приложение, которое связано с изменением состояния склада, после выполнения операций модификации данных, имитирующих продажу или удаление товара со склада, должно выполнять проверку на объем остатка, и в случае, если он меньше страхового запаса, формировать соответствующую заявку на поставку требуемого товара. Это усложняет клиентское приложение, с одной стороны, а с другой — может вызвать необоснованный заказ дополнительных товаров несколькими приложениями.

6. Сервер является активным, когда не только клиент, но и сам сервер, используя механизм триггеров, может быть инициатором обработки данных в БД.

7. Триггеры хранятся в словаре БД, они могут быть использованы несколькими клиентами, что существенно уменьшает дублирование алгоритмов обработки данных в разных клиентских приложениях.

8. Пользователю может быть разрешено выполнение над таблицей операций SELECT (ВЫБРАТЬ) и INSERT (ВКЛЮЧИТЬ).

Привилегии устанавливаются и отменяются специальными операторами языка SQL — GRANT (РАЗРЕШИТЬ) и REVOKE (ОТМЕНИТЬ).

Оператор GRANT указывает конкретного пользователя, который получает конкретные привилегии доступа к указанной таблице.

Например, оператор GRANT SELECT, INSERT ON Bills TO bit 123 устанавливает привилегии пользователю с логином bit 123 на выполнение операций выбора и включения над таблицей Bills. Как видно из примера, оператор GRANT устанавливает соответствие между операциями, пользователем и объектом базы данных (таблицей в данном случае).

9. Да

10. Здесь в систему добавляется новый слой, который размещается между клиентом и сервером, что увеличивает трату ресурсов на поддержку баланса загрузки актуальных серверов (load balancing) и ограничивает возможности управления взаимодействием «клиент–сервер». Во-первых, становится невозможным направить запрос от конкретного клиента конкретному серверу; во-вторых, серверы становятся равноправными — нет возможности устанавливать приоритеты для обслуживания запросов.