1. Приведите пример синхронизационного тупика. Как его определить?  
  
 а) транзакции  T1  и  T2  устанавливают  монопольные  блокировки  
объектов  o1  и  o2  
соответственно;  
 б)после  этого  T1  требуется  совместная  блокировка  объекта  o2,  
а  T2  -  совместная  
блокировка объекта o1;  
 в)ни  одно  из  этих  требований  блокировки  не  может  быть  
удовлетворено,  
следовательно, ни одна из транзакций не может продолжаться; поэтому  
монопольные  
блокировки объектов никогда не будут сняты, а требования совместных  
блокировок не  
будут удовлетворены.  
  
Основой  обнаружения  тупиковых  ситуаций  является  построение  (или  
постоянное  
поддержание)  графа  ожидания  транзакций.  Граф  ожидания  
транзакций  -  это  
ориентированный  двудольный  граф,  в  котором  существует  два  типа  
вершин  -  вершины,  
соответствующие транзакциям, и вершины, соответствующие объектам  
блокировок.  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 2. Приведите пример операции в БД, для которой не хватает блокировки  
картежей.  
  
Удаление таблицы. Надо блочить всю таблицу целиком  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 3. Какими свойствами должны обладать отношения, чтобы над ними можно  
было выполнить декартово произведение?  
  
 Два отношения совместимы по взятию расширенного декартова  
произведения в том и только в том случае, если пересечение множеств  
имен атрибутов, взятых из их схем отношений, пусто.  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 4. Какая основная проблема в операции проекции?  
  
 удаление дубликатов кортежей

------------------------------------------------------------------------------------------------------------------------------------------------  
 5. Назовите правило целостности сущности (из третей модели)  
  
Любой кортеж любого значения-отношения любой переменной отношения  
должен быть отличим от любого другого кортежа этого значения отношения  
по составным значениям заранее определенного множества атрибутов  
переменной отношения, т. е., другими словами, любая переменная  
отношения должна обладать первичным ключом.  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 6. Из чего следует, что отношения, атрибуты неупорядочены?  
Они определяются как множества.  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 7. Почему соединения не выполняется как прямое произведение?  
Потому что прямое произведение даёт слишком много кортежей и это медленно.  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 8. Приведите пример алгоритма операции соединения.  
/// Чую не то  
 В  общем  случае,  чтобы  получить  результат соединения общего вида  
произвольных отношений A и B, нужно:  
\* выполнить над одним из отношений одну или несколько операций  
<RENAME>, чтобы избавиться от общих имен атрибутов;  
\* выполнить  над  полученными  отношениями  операцию  <AND>,  
производящую расширенное декартово произведение;  
\* и  для  полученного  отношения  выполнить  одну  или  несколько  
операций  <AND> с отношениями-константами, чтобы должным образом  
ограничить его.  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 9. Почему для индексации БД используют B деревья?  
  
 Использование  B+-деревьев  позволяет  достичь  эффективности  при  
прямом  поиске,  поскольку  они  из-за  своей  сильной  ветвистости  
обладают  небольшой  
глубиной. Кроме того, B+-деревья сохраняют порядок ключей в листовых  
блоках иерархии, что позволяет производить последовательное  
сканирование таблицы в порядке возрастания  
или убывания значений полей, на которых определен индекс  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 10. Что такое связанные переменные?  
 Переменные, входящие в WFF, могут быть свободными или связанными.  
 Все переменные, входящие в WFF, при построении которой не  
использовались кванторы, являются свободными.  
 Фактически, это означает, что если для какого-то набора значений  
свободных кортежных переменных при вычислении WFF получено значение  
true,  
 то эти значения кортежных переменных могут входить в результирующее  
отношение.  
 Если же имя переменной использовано сразу после квантора при  
построении WFF вида EXISTS var (form) или FORALL var (form),  
 то в этой WFF и во всех WFF, построенных с ее участием, var - это  
связанная переменная. Это означает, что такая переменная не видна за  
пределами минимальной WFF, связавшей эту переменную.  
 При вычислении значения такой WFF используется не одно значение  
связанной переменной, а вся ее область определения.  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 11. Приведите пример правильно построенной формы формирования  
кортежей, которая не возвращает результат.

Если речь идет о WFF, то она ничего не может возвращать, так что  
вопрос странный.  
Я полагаю речь идет об операторе присваивания в алгебре Кодда, ибо он  
ничего не возвращает.  
И имеется в виду нечто такое  
Служащие := Служащие.  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 12. Как в SQL поддерживается авторизация пользователей?  
  
 Основная идея такого подхода состоит в том, что по отношению к любому  
отношению БД и любому столбцу отношения вводится предопределенный  
набор привилегий.  
 С каждой транзакцией неявно связывается идентификатор пользователя,  
от имени которого она выполняется (способы связи и идентификации  
пользователей не фиксируются в языке и определяются в реализации).  
  
 После создания нового отношения все привилегии, связанные с этим  
отношением и всеми его столбцами, принадлежат только пользователю-  
создателю отношения.  
 В число привилегий входит привилегия передачи всех или части  
привилегий другому пользователю, включая привилегию на передачу  
привилегий.  
 Технически передача привилегий осуществляется при выполнении  
оператора SQL GRANT. Существует также привилегия изъятия всех или  
части привилегий у пользователя, которому они ранее были переданы.  
 Эта привилегия также может передаваться. Технически изъятие  
привилегий происходит при выполнении оператора SQL REVOKE.  
  
 Проверка полномочности доступа к данным происходит на основе  
информации о полномочиях, существующих во время компиляции  
соответствующего оператора SQL.  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 13. Схема результата ограничения.  
  
Результатом ограничения отношения по некоторому условию является  
отношение, включающее кортежи отношения-операнда, удовлетворяющее  
этому условию.  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 14. Приведите пример отношения, в котором более одного ключа. В  
какой  нормальной форме находится это отношение?

Студент:  
Номер пасорта, Номер студака, Имя.  
  
В 1й. (Можно так постебаться над преподом, ибо все находится в 1й  
форме)  
Но по максимуму? вроде как в 5.  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 15. Приведите пример транзитивной формы.  
  
СЛУ\_НОМ->СЛУ\_ИМЯ->СЛУ\_ЗАРП (стр. 128, рис 7.3 слева)  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 16. Дайте определение второй нормальной формы.  
  
 Значения всех атрибутов отношения атомарны +  каждый  неключевой  
атрибут  (не  входящий  ни  в  один  возможный  ключ) минимально  
функционально зависит от первичного ключа  
 (предполагается, что у отношения имеется только один возможный  
ключ).  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 17. Ссылочное ограничение целостности.  
  
Для каждого значения внешнего ключа, появляющегося в кортеже  
значения-отношения ссылающейся переменной отношения, либо в  
значении-отношении переменной отношения, на которую указывает ссылка,  
должен найтись кортеж с таким же значением первичного ключа, либо  
значение внешнего ключа должно быть полностью неопределенным  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 18. Из какого свойства отношения следует, что у каждого отношения  
есть хотя бы один первичный ключ?  
  
 Уникальность кортежей(можно взять первичный ключ по всем атрибутам)  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 19. Какой особый предикат вводится при исчислении доменов?  
брр... Если R - это n-арное отношение с атрибутами a1, a2, ..., an, то  
условие членства имеет вид R (ai1 : vi1, ai2 : vi2, ..., aim : vim) (m  
n), где vij - это либо литерально задаваемая константа, либо имя  
доменной переменной. Условие членства принимает значение true в том и  
только в том случае, если в отношении R существует кортеж, содержащий  
указанные значения указанных атрибутов. Если vij - константа, то на  
атрибут aij накладывается жесткое условие, не зависящее от текущих  
значений доменных переменных; если же vij - имя доменной переменной,  
то условие членства может принимать разные значения при разных  
значениях этой переменной.  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 20. Что такое логически согласованное состояние БД?  
для Реляционой базы Лог. Согласованность <=> Целостность  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 21. Что такое физически согласованное состояние БД?  
  
 Моменты времени, в которые во внешней памяти содержатся согласованные  
результаты операций, завершившихся до соответствующего момента  
времени,  
 и отсутствуют результаты операций, которые не завершились, а буфер  
журнала вытолкнут во внешнюю память.  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 22. Что делается в иерархических системах, когда уничтожается запись  
предка?  
  
 В иерархической модели данных автоматически поддерживается  
целостность ссылок между  предками  и  потомками.  Основное  правило:  
никакой  потомок  не  может существовать  без  своего  родителя.  
\*Соответственно,все его потомки  
уничтожаются,не так ли?\*  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 23. Схема иерархической системы.  
  
 a) Структуры данных  
Иерархическая  БД  состоит  из  упорядоченного  набора  деревьев;  
более  точно,  из  
упорядоченного набора нескольких экземпляров одного типа дерева. Тип  
дерева состоит из  
одного  <<корневого>>  типа  записи  и  упорядоченного  набора  из  
нуля  или  более  типов  
поддеревьев  (каждое  из  которых  является  некоторым  типом  
дерева).  Тип  дерева  в  целом  
представляет собой иерархически организованный набор типов записи.  
  
Все  экземпляры  данного  типа  потомка  с  общим  экземпляром  типа  
предка  
называются  близнецами.  Для  иерархической  базы  данных  
определяется  полный  порядок  
обхода дерева: сверху-вниз, слева-направо.  
b) Манипулирование данными  
Примеры:  
\* найти указанный экземпляр типа дерева БД (например, отдел 310);  
\* перейти от одного экземпляра типа дерева к другому;  
\* перейти от экземпляра одного типа записи к экземпляру другого типа  
записи внутри  
дерева (например, перейти от отдела к первому сотруднику);  
\* перейти от одной записи к другой в порядке обхода иерархии;  
\* вставить новую запись в указанную позицию;  
\* удалить текущую запись.  
c) Ограничения целостности  
В иерархической модели данных автоматически поддерживается целостность  
ссылок  
между  предками  и  потомками.  Основное  правило:  никакой  потомок  
не  может  
существовать  без  своего  родителя.  Заметим,  что  аналогичная  
поддержка  целостности  по  
ссылкам между записями без связи <<предок-потомок>>, не обеспечивается.  
Примером такой  
<<внешней>>  ссылки  является  содержимое  поля  Рук\_Отдел  в  
экземпляре  типа  записи Руководитель.  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 24. Что такое тип связи в сетевых БД?  
 Тип связи определяется для двух типов записи: предка и потомка.  
Экземпляр типа связи  
состоит из одного экземпляра типа записи предка и упорядоченного  
набора экземпляров  
типа  записи  потомка.  Для  данного  типа  связи L с  типом записи  
предка  P  и  типом записи  
потомка C должны выполняться следующие два условия:  
\* каждый экземпляр типа записи P является предком только в одном  
экземпляре типа  
связи L;  
\* каждый  экземпляр  типа  записи  C  является  потомком  не  более  
чем  в  одном  
экземпляре типа связи L.  
На  формирование  типов  связи  не  накладываются  особые  
ограничения;  возможны,  
например, следующие ситуации:  
\* тип записи потомка в одном типе связи L1 может быть типом записи  
предка в другом  
типе связи L2 (как в иерархии);  
\* данный тип записи P может быть типом записи предка в любом числе  
типов связи;  
\* данный тип записи P может быть типом записи потомка в любом числе  
типов связи;  
\* может  существовать  любое  число  типов  связи  с  одним  и  тем  
же  типом  записи  
предка и одним и тем же типом записи потомка; и если L1 и L2 - два  
типа связи с одним  
и тем же типом записи предка P и одним и тем же типом записи потомка  
C, то правила,  
по которым образуется родство, в разных связях могут различаться;  
\* типы  записи  X  и  Y  могут  быть  предком  и  потомком  в  одной  
связи  и  потомком  и  
предком - в другой;  
\* предок и потомок могут быть одного типа записи.  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 25. Почему для индексации не используются двоичные деревья?  
  
Их высота выше, а информации в узле значительно меньше, чем размер  
буфера, который ОС читает с диска. Т.е. выгоднее, чтобы размер узла  
был большой (примерно равен размеру буфера чтения), а высота (от неё  
зависит кол-во чтений) поменьше, что и обеспечивают B+ деревья.  
  
------------------------------------------------------------------------------------------------------------------------------------------------  
 26. Для какого компонента СУБД важна балансировка дерева?  
  
Для RSS, очевидно же. Нет надобности париться о том, что  
страничка, где была ветка переполнилась. Ветка при переполнении сама  
все уладит.