**1. Дайте определение понятию «база данных» (БД).**

База данных (БД) — это организованная структура, предназначенная для хранения, изменения и обработки взаимосвязанной информации, преимущественно больших объемов.

**2. Дайте определение понятию «система управления базой данных» (СУБД).**

Система управления базами данных (СУБД) — это комплекс программных средств, необходимых для создания структуры новой базы, ее наполнения, редактирования содержимого и отображения информации.

**3. Назовите основные компоненты СУБД.**

* **Ядро**, которое отвечает за управление данными во внешней и оперативной памяти и журнализацию,
* **Процессор языка базы данных**, обеспечивающий оптимизацию запросов на извлечение и изменение данных и создание, как правило, машинно-независимого исполняемого внутреннего кода,
* **Подсистему поддержки времени исполнения**, которая интерпретирует программы манипуляции данными, создающие пользовательский интерфейс с СУБД
* **Сервисные программы** (внешние утилиты), обеспечивающие ряд дополнительных возможностей по обслуживанию информационной системы.

**4.**

**5. Какие БД называются реляционными? Назовите основные признаки реляционных баз данных.**

**Реляционная база данных** представляет собой множество взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного вида. Каждая строка таблицы содержит данные об одном объекте (например, автомобиле, компьютере, клиенте), а столбцы таблицы содержат различные характеристики этих объектов - атрибуты (например, номер двигателя, марка процессора, телефоны фирм или клиентов).

В зависимости от структуры данных различают иерархические, сетевые и реляционные базы данных.

**Реляционной считается такая база данных**, в которой все данные представлены в виде двумерных таблиц и все операции над базой сводятся к манипуляциям над таблицами. Название «реляционная» связано с тем фактом, что каждая запись в такой базе содержит информацию, относящуюся (related) только к одному конкретному объекту (экземпляру сущности). Кроме того, с данными двух сущностей можно работать как с единым целым, основанным на значениях связей (ключей) между сущностями.

**6. Что такое *OLTP*?**

OLTP (Online Transaction Processing)

**OLTP** (обработка транзакций в режиме реального времени) участвует в работе конкретной системы. OLTP характеризуется большим количеством коротких онлайновых транзакций (INSERT, UPDATE, DELETE). Основной упор для OLTP-систем заключается в очень быстрой обработке запросов, обеспечении целостности данных в средах с множественным доступом и эффективности, измеряемой количеством транзакций в секунду. В базе данных OLTP есть подробные и текущие данные, а схема, используемая для хранения транзакционных баз данных, - это модель сущности (обычно 3NF). Он включает в себя Запросы, связанные с индивидуальной записью, например "Обновление электронной почты" в базе данных компании.

**7. Что такое *OLAP*?**

(online analytical processing)

**OLAP** (онлайн аналитическая обработка) имеет дело с историческими данными или архивными данными. OLAP характеризуется относительно низким объемом транзакций. Запросы часто очень сложны и включают скопления. Для систем OLAP время отклика - это показатель эффективности. Приложения OLAP широко используются методами интеллектуального анализа данных. В базе данных OLAP хранятся агрегированные исторические данные, хранящиеся в многомерных схемах (обычно звездообразная схема). Иногда запрос должен получить доступ к большому количеству данных в записях управления, как то, что было прибылью вашей компании в прошлом году.

**8. Поясните понятие «архитектура клиент-сервер».**

**«Клиент — сервер»** — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами. Фактически клиент и сервер — это программное обеспечение. Обычно эти программы расположены на разных вычислительных машинах и взаимодействуют между собой через вычислительную сеть посредством сетевых протоколов, но они могут быть расположены также и на одной машине.

**9. Поясните понятие «сервис Windows».**

**Сервис, или служба Windows,** - это фоновое приложение, которое может запускаться различными способами, в том числе автоматически при старте Windows, или стартовать в том случае, если окажется нужным другому подобному приложению. Отличительной особенностью сервиса является отсутствие средств непосредственного диалога с пользователем и, соответственно, отсутствие необходимости для пользователя предпринимать какие-либо действия для поддержания сервиса в работоспособном состоянии. Большинство сервисов могут быть остановлены по запросу пользователя, однако есть и такие, которые не могут быть остановлены по причине того, что они необходимы для нормальной работы некоторых компонентов операционной системы.

**10. Поясните понятие «программный интерфейс СУБД».**

**API** (**программный интерфейс приложения, интерфейс прикладного программирования**) (***a****pplication****p****rogramming****i****nterface*) — набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) или операционной системой для использования во внешних программных продуктах.

Существует два основных способа доступа к данным из клиентских приложений: использование прикладного интерфейса и использование универсального программного интерфейса.

*Прикладной* программный интерфейс (API – Application Programming Interface) представляет собой набор функций, вызываемых из клиентского приложения. Он может работать только с СУБД данного производителя и при ее замене придется переписывать значительную часть кода клиентского приложения. Прикладной программный интерфейс различен для разных СУБД.

*Универсальный* механизм доступа к данным обеспечивает возможность использования одного и того же интерфейса для доступа к разным типам СУБД. Обычно он реализован в виде специальных дополнительных модулей, называемых драйверами.

Наиболее распространенным программным интерфейсом, обеспечивающим доступ к данным конкретной базы данных является ODBC (Open Database Connectivity) фирмы Microsoft. В рамках ODBC программное приложение непосредственно взаимодействует с диспетчером драйвером, посылая ему ODBC-вызовы. Диспетчер драйверов отвечает за динамическую загрузку нужного ODBC-драйвера, через который обращается с серверу баз данных. ODBC-драйвер выполняет все вызовы ODBC-функций и «переводит» их на язык источника данных. СУБД хранит и выводит данные в ответ на запросы со стороны ODBC-драйвера.

**11. Поясните понятие «интерфейс командной строки»**

**Интерфейс командной строки** (Command lineinterface, CLI) — разновидность текстового интерфейса(CUI) между человеком и компьютером, в котором инструкции компьютеру даются в основном путём ввода с клавиатуры текстовых строк (команд).

**Текстовый пользовательский интерфейс**, (*Text user interface, TUI*; также *Character User Interface, CUI*) — разновидность интерфейса пользователя, использующая при вводе-выводе и представлении информации исключительно набор буквенно-цифровых символов и символов псевдографики.

**12. Расшифруйте SQL и объясните, что это такое.**

**SQL** ( *structured query language* — «язык структурированных запросов») — декларативный язык программированияя, применяемый для создания, модификации и управления данными в реляционной базы данных управляемой соответствующей СУБД.

**13. Поясните понятие «декларативный язык». Приведите примеры декларативных языков.**

**Декларативное программирование** - это когда вы пишете свой код таким образом, чтобы он описывал, что вы хотите делать, а не как вы хотите это сделать. Остается до компилятора выяснить, как это сделать.

SQL & Prolog.

**14. Поясните понятие «процедурный язык». Приведите примеры процедурных языков.**

**Процедурный язык программирования** предоставляет возможность программисту определять каждый шагв процессе решения задачи. Особенность таких языков программирования состоит в том, что задачиразбиваются на шаги и решаются шаг за шагом. Используя процедурный язык, программист определяетязыковые конструкции для выполнения последовательности алгоритмических шагов.

Программа на процедурном языке программирования состоит из последовательности операторов(инструкций), задающих процедуру решения задачи.Выполнение программы сводится к последовательномувыполнению операторов с целью преобразования исходного состояния памяти, то есть значений исходныхданных, в заключительное, то есть в результаты.

ADA & Basic & Fortan & Pascal

**15-16. Перечислите группы операторов языка SQL.** **Назовите основные операторы каждой группы.**

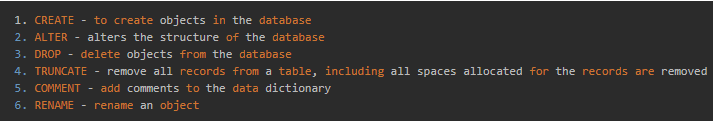
Язык определения данных DDL (Data Definition Language);(CREATE DROP ALTER)

Язык манипулирования данными DML (Data Manipulation Language).

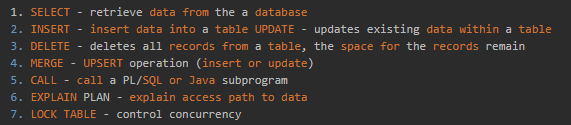
Язык определения данных включает операторы, управляющие объектами базы данных. К последним относятся таблицы, индексы, представления. Для каждой конкретной базы данных существует свой набор объектов базы данных, который может значительно расширять набор объектов, предусмотренный стандартом. В некоторых СУБД, таких как Oracle, все объекты базы данных, принадлежащие одному пользователю, образуют схему базы данных. С другой стороны, в стандарте SQL92 термином "схема" стали называть группу взаимосвязанных таблиц.

Язык манипулирования данными включает операторы, управляющие содержанием таблиц базы данных и извлекающими информацию из этих таблиц.

DLL:



Язык DML определяет следующие операторы:



* SELECT - извлечение данных из одной или нескольких таблиц;
* INSERT - добавление строк в таблицу;
* DELETE - удаление строк из таблицы;
* UPDATE - изменение значений полей в таблице.

**17. Поясните понятие «ограничения целостности». Перечислите все известные вам типы ограничений целостности**

***Ограничение целостности*** - это некоторое утверждение, которое может быть истинным или ложным в зависимости от состояния базы данных.

**Ограничения могут быть внутренними (неявными) и явными**.

**Внутренние ограничения**предусматриваются самой моделью данных и тесно связаны со структурой данных. С ограничениями этого типа хорошо согласуются операции манипулирования данными, поэтому контроль над соблюдением внутренних ограничений обычно не вызывает трудностей. Как правило, внутренние ограничения целостности задаются вместе с определением структурных спецификаций средствами языка описания данных (ЯОД - Data Definition Language(DDL)).

**Явные ограничения** включаются в структуру базы данных с помощью средств языка контроля данных (DCL, Data Control Language). В качестве явных ограничений чаще всего выступают условия, накладываемые на значения данных. Например, номер паспорта является уникальным, заработная плата не может быть отрицательной, а дата приёма сотрудника на работу обязательно будет меньше, чем дата его перевода на другую работу.

**К внутренним ограничениям целостности можно отнести:**

1) ограничения на значения ключевых атрибутов: уникальность значений атрибутов первичных и альтернативных ключей определяется понятием **ключа**. Эти ограничения задаются соответствующим выделением ключей в концептуальной схеме (PRIMARY KEY, UNIQUE);

2) ограничения на допустимые значения атрибутов: все допустимые значения атрибутов должны удовлетворять условию принадлежности соответствующему домену. Эти ограничения задаются соответствующим описанием доменов и атрибутов. Кроме того, в описании атрибутов указывается обязательность значений определенных атрибутов (NULL, NOT NULL);

3) ограничения на существующие значения (ссылочные ограничения): существование одних сущностей (дочерних, типа категория) ставится в зависимость от существования других (родительских, родовых). Эти ограничения представляются с помощью соответствующих связей между множествами сущностей.

**Явные ограничения целостности задаются разработчиками.**

**18. Перечислите все секции оператора SELECT в порядке их выполнения. Поясните назначение каждой секции.**

* SELECT определяет список возвращаемых столбцов (как существующих, так и вычисляемых), их имена, ограничения на уникальность строк в возвращаемом наборе, ограничения на количество строк в возвращаемом наборе;
* FROM задаёт табличное выражение, которое определяет базовый набор данных для применения операций, определяемых в других предложениях оператора;
* WHERE задает ограничение на строки табличного выражения из предложения FROM;
* GROUP BY объединяет ряды, имеющие одинаковое свойство с применением агрегатных функций
* HAVING выбирает среди групп, определенных параметром GROUP BY
* ORDER BY задает критерии сортировки строк; отсортированные строки передаются в точку вызова.

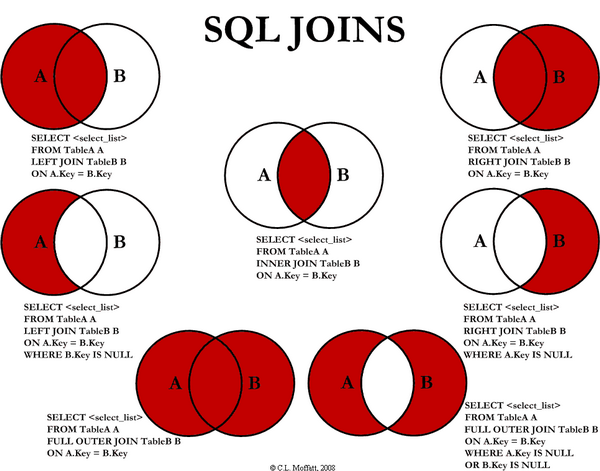
Поярдок выполнения SELECT:

* FROM.
* ON.
* JOIN.
* where.
* GROUP BY.
* WITH CUBE или WITH ROLLUP.
* HAVING.
* **SELECT**.

**19. Перечислите все способы соединения таблиц в SELECT-запросе? Назовите коммутативные способы соединения таблиц.**

Результирующий набор создается следующим образом: выполняется декартово произведение (каждая строка одной таблицы соединяется с каждой строкой другой).

Из полученного результата выбираются строки, удовлетворяющие указанному условию; из всех столбцов результирующего набора выбираются столбцы, указанные в списке SELECT.



**20. Поясните понятия «подзапрос», «некоррелируемый подзапрос» и «коррелируемый подзапрос».**

**Подзапрос** – это SELECT-запрос, который выполняется в рамках другого запроса. Подзапросы могут применяться в секции WHERE. Подзапросы бывают двух видов: коррелируемые и независимые. *Коррелируемый* подзапрос зависит от внешнего запроса и выполняется для каждой строки результирующего набора. *Независимый* подзапрос не зависит от внешнего запроса и выполняется только один раз, но результат его выполнения подставляется в каждую строку результирующего набора. В SELECT-списке допускается применять только такие подзапросы, которые формируют скалярный результирующий набор (набор, состоящий из одной строки и одного столбца).

**21. Поясните понятие «агрегатная функция». Перечислите известные вам агрегатные функции.**

Агрегатные функции выполняют вычисление на наборе значений и возвращают одиночное значение. Агрегатные функции, за исключением COUNT, не учитывают значения NULL.

**AVG** (вычисление среднего значения), **COUNT** (вычисление количества строк), **MAX**(вычисление максимального значения), **MIN** (вычисление минимального значения), **SUM** (вычисление суммы значений).

При использовании секции **GROUP BY** в SELECT-списке допускается указывать только те столбцы, по которым осуществляется группировка.

**22. Поясните применение конструкций IN, ALL, ANY, BETWEEN, NОТ, IS NULL, LIKE, EXISTS в секции WHERE.**

Операция IN формирует логическое значение «истина» в том случае, если значение, указанное слева от ключевого слова IN, равно хотя бы одному из значений списка, указанного справа от IN.

Операция >ALL сформирует истинное значение в том случае, если значение, стоящее слева, больше каждого значения в списке, указанном справа.

При выполнении ANY-сравнения сравнивается (операция >=) с каждым значением списка, записанным справа . Если хотя бы для одного значения из списка операция сравнения принимает значение «истина», то вся операция (>= ANY) принимает значение «истина».

BETWEEN, по своей сути это выражение проверки на принадлежность к диапазону значений.

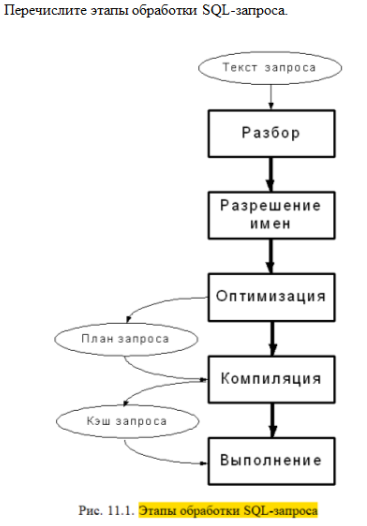
Оператор NOT отображает запись, если условие (и) НЕ ИСТИНА.

NULL позволяет проверить отсутствие (наличие) значения в полях таблицы.

операция LIKE проверяет вхождение подстроки в строку

EXISTS Результатом операции является «истина», если результирующий набор подзапроса содержит хотя бы одну строку, в противоположном случае операция EXISTS формирует значение «ложь».

**23.**



**24. Поясните понятия «план запроса»-** План запроса представляет собой алгоритм выполнения SQL-запроса., «стоимость запроса» - (время) оценивается затратами ресурсов компонент системы, называемыми стоимостью выполнения запроса.

**25.** Создание индекса является методом увеличения производительности работы СУБД при извлечению записей. В индексе создается запись для каждого значения, которое появляется в индексируемом столбце. По умочанию, Oracle создает индексы типа B-дерево.

**26.** Уникальные и неуникальные индексы. Уникальные индексы основаны на уникальном столбце — обычно вроде номера карточки социального страхования сотрудника. Хотя уникальные индексы можно создавать явно, Oracle не рекомендует это делать. Вместо этого следует использовать уникальные ограничения. Когда накладывается ограничение уникальности на столбец таблицы, Oracle автоматически создает уникальные индексы по этим столбцам.

\* Первичные и вторичные индексы. Первичные индексы — это уникальные индексы в таблице, которые всегда должны иметь какое-то значение и не могут быть равны null. Вторичные индексы — это прочие индексы таблицы, которые могут и не быть уникальными.

\* Составные индексы. Составные индексы — это индексы, содержащие два или более столбца из одной и той же таблицы. Они также известны как сцепленные индексы (concatenated index). Составные индексы особенно полезны для обеспечения уникальности сочетания столбцов таблицы в тех случаях, когда нет уникального столбца, однозначно идентифицирующего строку.

**27.** Oracle автоматически создает уникальный индекс по уникальному или первичному ключу. База данных создаст индекс автоматически, когда будет включено ограничение, и по умолчанию он получит имя соответствующего ограничения.

**28.** Поясните понятие «фрагментация индекса».

Операции добавления и изменения строк БД могут повлечь образование неиспользуемых фрагментов в области памяти индекса. Индексы таблиц становятся пористыми (фрагментированными). Процесс образования

неиспользуемых фрагментов памяти называют фрагментацией. Фрагментация индексов снижает эффект от их применения.

**29.** Сколько у одной таблицы может быть кластеризованных индексов и почему?

Кластеризованный индекс хранит реальные строки данных в листьях индекса. Важной характеристикой кластеризованного индекса является то, что все значения отсортированы в определенном порядке либо возрастания, либо убывания. Таким образом, таблица или представление может иметь только один кластеризованный индекс

**30.** Поясните назначение процедур перестройки и реорганизации индексов. В чем разница?

Реорганизация индекса вызывает переупорядочивание данных в наиболее удаленных страницах данных и уплотняет индекс. Никакие дополнительные данные не добавляются в индекс при реорганизации, поэтому индекс может остаться несколько фрагментированным. Операция не потребляет много ресурсов системы и может происходить, когда внешние процессы получают доступ к таблице, на которой работает этот индекс. Поэтому мы говорим в этом случае об "онлайновой' операции.

Перестройка индекса по существу уничтожает индекс и создает новый. Любая фрагментация, которая была в старом индексе, устраняется, и логический порядок нового индекса соответствует физическому порядку. Вследствие того, что индекс удаляется и создается вновь, внешние процессы не в состоянии получить доступ к таблице, что отрицательно сказывается на производительности. Фактически, другие процессы не будут в состоянии вообще заблокировать таблицу, пока идет процесс перестройки индекса. Это - главный недостаток перестройки индексов.

31. Поясните понятие «представление». С помощью какого оператора создается представление?

Представление (View) – это объект базы данных, представляющий собой поименованный SELECT-запрос, который хранится в базе данных. Представление создается с помощью оператора CREATE, удаляется с помощью оператора DROP и изменяется с помощью ALTER.

32. В каких случаях к представлению применимы операторы INSERT, DELETE, UPDATE?

При создании представлений, позволяющих выполнять операции INSERT, DELETE и UPDATE, базовый SELECT-запрос должен удовлетворять правилам:

\* запрос не должен содержать секцию группировки GROUP BY;

\* запрос не должен применять агрегатные функции, опции DISTINCT и TOP, операторы UNION, INTERSECT и EXCEPT;

\* в SELECT-списке запроса не должно быть вычисляемых значений;

\* в секции FROM запроса должна указываться только одна таблица.

33. Какое дополнительное свойство приобретает представление с опцией WITH CHECK OPTION?

Предложение WITH CHECK OPTION определяет, что INSERT-ы и UPDATE-ы, выполняемые посредством представления, не могут создавать строки, которые представление не может выбрать. Если происходит попытка выполнить операции DML на строках, которые не может выбрать представление, ошибка выводится на экран, наряду с именем ограничения, если оно было определено.

34.Команда DELETE удаляет записи из таблицы, которые удовлетворяют критерию WHERE.

TRUNCATE удаляет все данные из таблицы.

TRUNCATE быстрее и использует меньше системных ресурсов, чем DELETE и практически не пишет лог транзакции.

TRUNCATE удаляет данные путем деаллокации тех страниц, которые хранят табличные данные и только эти операции деаллокации записываются в лог транзакции.

TRUNCATE удаляет все строки таблицы, но структура таблицы (столбцы, ограничения, индексы и т.д.) остается. Счетчик, который используется для уникальности новых записей обнуляется.

Нельзя использовать TRUNCATE TABLE для таблиц, связанных ограничением FOREIGN KEY.

Откат (rollback) после TRUNCATE невозможен

TRUNCATE - команда DDL

DELETE удаляет строки в таблицы и для каждой оставляет запись в логе транзакции.

DELETE не обнуляет счетчик уникальности.

DELETE может использоваться с выражением WHERE или без него

DELETE активирует триггеры

После DELETE возможен откат

DELETE - команда DML

35.Под курсором в Oracle понимается получаемый при выполнении запроса результирующий набор и связанный с ним указатель текущей записи.

\* Неявные курсоры - простой оператор SELECT ... INTO извлекает одну строку данных непосредственно в переменные локальной программы. Это удобный (и часто наиболее эффективный) способ доступа к данным, использование которого, однако, может приводить к необходимости повторного кодирования оператора SELECT (или похожих операторов) в нескольких местах программы.

\* Явные курсоры - вы можете явно объявить курсов в разделе объявлений (локального блока или пакета). В этом случае курсор можно будет открывать и извлекать данные в одной или нескольких программах, причем возможности контроля будут шире, чем при использовании неявных курсоров.

\* Курсорные переменные - дополнительный уровень гибкости обеспечивают курсорные переменные (объявленные на основе типа REF CURSOR), которые позволяют передавать указатель на результирующее множество, полученное по запросу из одной программы в другую. Любая программа, имеющая доступ к такой переменной, сможет открывать и закрывать курсор, а также выбирать из него данные.

\* Курсорные выражения - появившиеся в версии Oracle 9i выражения CURSOR преобразуют оператор SELECT в указатель (типа REF CURSOR) на результирующее множество и могут использоваться в сочетании с табличными функциями для повышения прозводительности приложений.

Явный курсор объявляется разработчиком, а неявный курсор не требует объявления.

36.В соответствии со стандартом SQL при работе с курсорами можно выделить следующие основные действия:

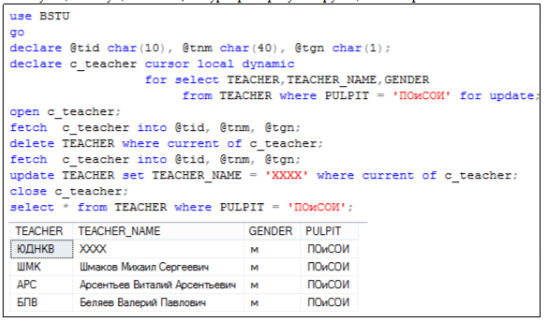
создание или объявление курсора;

открытие курсора , т.е. наполнение его данными, которые сохраняются в многоуровневой памяти;

выборка из курсора и изменение с его помощью строк данных;

закрытие курсора, после чего он становится недоступным для пользовательских программ;

освобождение курсора, т.е. удаление курсора как объекта, поскольку его закрытие необязательно освобождает ассоциированную с ним память.

37. В секции WHERE операторы DELETE и UPDATE (рис. 13.38) используют операцию CURRENT OF, для которой указывается имя курсора (в примере C\_TEACHER). Такой формат операторов позволяет удалять или изменять строки в таблице (TEACHER), соответствующие текущей позиции курсора в результирующем наборе. 

38. Транзакция – это несколько DML-операторов, которые либо все успешно выполнятся, либо все не выполняются. TCL SQL включает четыре оператора: BEGIN TRAN, SAVE TRAN, COMMIT TRAN и ROLLBACK TRAN.

Транзакция в первом приближении – это набор операторов DML SQL, которые рассматриваются как единое целое и могут либо все полностью выполниться (и изменить БД) или все не выполниться (и БД не изменится), причем ключевое слово здесь «все».

Любая транзакция завершается либо фиксацией (оператор COMMIT), либо откатом (оператор ROLLBACK).

Откат – это отмена всех изменений в БД, выполненных в рамках транзакции. Фиксация – подтверждение успешного завершения транзакции.

39. Обычно при определении понятия транзакции используют перечень требований к этому механизму, сформулированный в конце 70-х гг. Джимом Греем в виде аббревиатуры ACID: Atomicity – атомарность, Consistency – согласованность, Isolation – изолированность, Durability – долговечность.

Атомарность транзакции заключается в гарантии того, что никакая транзакция не будет зафиксирована в БД частично: операторы изменения БД, включенные в транзакцию, либо выполнятся все, либо не выполнится ни один. В БД это свойство реализуется с помощью механизма отката, позволяющего отменить выполненные, но незафиксированные изменения БД.

Свойство согласованности заключатся в том, что транзакция должна фиксировать новое согласованное состояние БД. Понятие согласованности зависит от предметной области, состояние которой отражается в БД.

Современные СУБД допускают одновременную (часто говорят – параллельную) работу нескольких транзакций. При этом параллельные транзакции могут изменять одни и те же данные в БД. Свойство изолированности гарантирует с точностью до определенного уровня отсутствие взаимного влияния параллельных транзакций на результаты их выполнения. Современные СУБД могут обеспечивать несколько уровней изолированности

транзакций. Уровни изолированности транзакций, поддерживаемые MSS, будут рассматриваться позже.

Любая транзакция завершается либо фиксацией (оператор COMMIT), либо откатом (оператор ROLLBACK). Откат – это отмена всех изменений в БД, выполненных в рамках транзакции. Фиксация – подтверждение успешного завершения транзакции. Свойство долговечности заключается в том, что изменения в БД, выполненные и зафиксированные транзакцией, не могут быть отменены, кроме как с помощью новой транзакции.

**40) Поясните понятия «режим автофиксации», «режим неявной транзакции», «режим явной транзакции»**

**Режим автоматической фиксации (часто просто автофиксация) транзакции используется MSS по умолчанию. После завершения каждый оператор T-SQL фиксируется или откатывается. Если оператор выполняется без ошибок, то автоматически выполняется фиксация. В противном случае состояние БД откатывается назад. Действие режима автофиксации продолжается до тех пор, пока не запускается явная транзакция или не включается режим неявной транзакции. Если явная транзакция завершается (фиксируется или откатывается) или выключается режим неявной транзакции, то автоматически осуществляется возврат в режим автофиксации.**

**Неявная транзакция продолжается до тех пор, пока не будет выполнен оператор фиксации (COMMIT) или оператор отката (ROLLBACK) транзакции. Режим неявной транзакции может быть включен для текущего соединения с сервером БД с помощью специальной инструкции T-SQL: SET IMPLICIT\_TRANSACTIONS ON. После выполнения этой инструкции соединение переключается из режима автофиксации в режим неявной транзакции. Обратное переключение осуществляется с помощью аналогичной инструкции, но с использованием ключевого слова OFF вместо ON.**

**Переключение в режим явной транзакции осуществляется с помощью оператора BEGIN TRANSACTION. Транзакцию должен завершать один из операторов: COMMIT TRANSACTION или ROLLBACK TRANSACTION. Ключевое слово TRANSACTION в перечисленных операторах может быть сокращено до TRAN. После завершения явной транзакции происходит возврат в исходный режим (автофиксации или неявной транзакции).**

**--------------------------------------**

**41) Перечислите известные вам уровни изолированности в порядке их усиления**

**Под «уровнем изоляции транзакций» понимается степень обеспечиваемой внутренними механизмами СУБД (то есть не требующей специального программирования) защиты от всех или некоторых видов вышеперечисленных несогласованностей данных, возникающих при параллельном выполнении транзакций.**

**Стандарт определяет шкалу из четырёх уровней изоляции: Read uncommitted, Read committed, Repeatable read, Serializable. Первый из них является самым слабым, последний — самым сильным, каждый последующий включает в себя все предыдущие.**

**Уровень изоляции Фантомное Неповторяющееся «Грязное» Потерянное**

**SERIALIZABLE + + + +**

**REPEATABLE READ - + + +**

**READ COMMITTED - - + +**

**READ UNCOMMITTED - - - +**

**NULL - - - -**

**----------------------------------------**

**42) Поясните понятие «неподтвержденное чтение», «неповторяющееся чтение», «фантомное чтение»**

**«грязное» чтение (англ. dirty read) — чтение данных, добавленных или изменённых транзакцией, которая впоследствии не подтвердится (откатится);**

**неповторяющееся чтение (англ. non-repeatable read) — при повторном чтении в рамках одной транзакции ранее прочитанные данные оказываются изменёнными;**

**фантомное чтение (англ. phantom reads) — одна транзакция в ходе своего выполнения несколько раз выбирает множество строк по одним и тем же критериям. Другая транзакция в интервалах между этими выборками добавляет или удаляет строки или изменяет столбцы некоторых строк, используемых в критериях выборки первой транзакции, и успешно заканчивается. В результате получится, что одни и те же выборки в первой транзакции дают разные множества строк.**

43. Поясните понятия «хранимая процедура», «входной параметр», «выходной параметр», «значение, возвращаемое к точке вызова», «позиционная форма передачи параметров», «параметрическая форма передачи параметров», «системная хранимая процедура».

Хранимая процедура – это объект базы данных, представляющий собой поименованный код T-SQL. Хранимая процедура может быть создана с помощью CREATE, изменена с помощью ALTER и удалена с помощью оператора DROP. Процедура может принимать входные и формировать выходные параметры. Результатом ее выполнения может быть целочисленное значение, которое возвращается к точке вызова оператором RETURN, либо один или более результирующих наборов, сформированных операторами SELECT, либо содержимое стандартного выходного потока, полученного при выполнении операторов PRINT. Вызов процедуры осуществляется оператором EXECUTE (EXEC).

В хранимых процедурах допускается применение основных DDL и всех DML и TCL-операторов, конструкций TRY/CATCH, курсоров, временных таблиц.

Поскольку мы говорим о Microsoft SQL Server, следует отметить огромное количество системных хранимых процедур, реализованных в нем. Имена системных хранимых процедур начинаются с SP\_ или XP\_ и хранятся в базе данных master. Выше мы уже описывали некоторые из часто используемых системных хранимых процедур.

Системные хранимые процедуры можно использовать в клиентских приложениях. Для этого следует обратиться к базе данных master и выбрать имя соответствующей процедуры.

sp\_configure Используется для вывода или изменения глобальных установок конфигурации сервера

sp\_databases Используется для получения списка баз данных, имеющихся на сервере или доступных с его помощью

sp\_datatype\_info Используется для получения сведений о поддерживаемых типах данных

sp\_help Используется для получения сведений об объекте в базе данных

sp\_helpdb Используется для получения сведений об указанной базе данных или о всех базах данных, доступных с помощью данного сервера

sp\_helpfile Используется для получения сведений о физических именах и атрибутах файлов, в которых содержится текущая база данных

sp\_monitor Выводит статистику Microsoft SQL Server

44. Поясните понятия «скалярная функция», «встроенная табличная функция», «многооператорная хранимая функция».

Функция – это объект БД, представляющий собой поименованный код T-SQL. Для создания, удаления и изменения функций надо использовать операторы CREATE, DROP и ALTER соответственно.

Результатом выполнения функции является возвращаемое к точке вызова значение. Если функция возвращает единственное значение (число, строка, дата, время и пр.), то она называется скалярной. Функция, возвращающая таблицу, называется табличной. В зависимости от структуры кода, различают встроенные функции и многооператорные табличные функции.

45. Назовите отличия хранимых процедур от функций.

Отличие функций от хранимых процедур в ограничениях, накладываемых на код функции, в форме представления результата работы, а также в способе вызова. В функции не допускается применение DDL-операторов, DML-операторов, изменяющих БД (INSERT, DELETE, UPDATE), конструкций TRY/CATCH, а также использование транзакций.