Лабораторная работа № 3 - АБДиП

Ответьте на следующие вопросы:

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

БД – совокупность данных, систематизированных таким образом, чтобы эти данные могли храниться и обрабатываться ЭВМ.

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

СУБД – это совокупность программных средств, обеспечивающих управление созданием и использованием баз данных.

1. **Назовите основные компоненты СУБД.**

Ядро (управдение БД во внешней и оперативной памяти)

Процессор языка БД (оптимизация запросов на изменение и извлечение и создание машинно-независимого исполняемого кода)

Подсистему поддержки времени исполнения ()

Сервисные программы

1. **Какие БД называют системными? y**

Любая реляционная СУБД, поддерживающая списки пользователей и ролей, должна их где-то хранить. В дополнение к этим спискам многие СУБД хранят списки таблиц, индексов, триггеров, процедур и др., а также сведения о том, кто ими владеет. Эти списки называются системными таблицами (system tables), а соответствующая часть базы данных называется системным каталогом (system catalog).

Отметим, что не все СУБД поддерживают системные каталоги (например, их не поддерживают такие популярные в недавнем прошлом СУБД, как dBase и Paradox).

1. **Какие БД называются реляционными? Назовите основные признаки реляционных баз данных.**
2. **Что такое *OLTP*? Y**

Обработка транзакций в реальном времени. Способ организации БД при котором система работает с небольшими транзакциями, которые идут большим потоком при этом нужно минимальное время отклика от системы.

1. **Что такое *OLAP*?**

Обработка запросов в реальном времени с динамическим составлением отчетов

1. **Поясните понятие «архитектура клиент-сервер».**
2. **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.

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

программная компонента, обеспечивающая соединение с сервером БД, выполнение команд и получение результатов их выполнения

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

 разновидность текстового **интерфейса** (TUI) между человеком и компьютером, в котором инструкции компьютеру даются в основном путём ввода с клавиатуры текстовых **строк**(команд)

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

 «язык структурированных запросов») — [формальный](https://ru.wikipedia.org/wiki/%D0%A4%D0%BE%D1%80%D0%BC%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) непроцедурный язык программирования, применяемый для создания, модификации и управления данными в произвольной [реляционной базе данных](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), управляемой соответствующей системой управления базами данных ([СУБД](https://ru.wikipedia.org/wiki/%D0%A1%D0%A3%D0%91%D0%94)). SQL основывается на [исчислении кортежей](https://ru.wikipedia.org/wiki/%D0%98%D1%81%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BA%D0%BE%D1%80%D1%82%D0%B5%D0%B6%D0%B5%D0%B9).

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

*Декларативное программирование*. Описывает процесс каково нечто, а не как его создать. Порядок выполнения и способ достижения не важен.

Примером декларативного языка служит HTML, описывающий содержание страницы, а не способ её отображения на экране. Популярный сборщик пакетов Maven описывает лишь зависимости между пакетами, а не последовательность их установки. Язык запросов SQL описывает, что мы хотим достать из базы данных, как именно решает сервер. **Декларативное описание задачи более наглядно и легче формулируется, так как мы чаще знаем, чего хотим, но не знаем как сделать.**

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

Выполнение программы сводится к последовательному выполнению операторов с целью преобразования исходного состояния памяти, то есть значений исходных данных, в заключительное, то есть в результаты. Таким образом, с точки зрения программиста имеются программа и память, причем первая последовательно обновляет содержимое последней.

**Процедурный язык программирования** предоставляет возможность программисту определять каждый шаг в процессе решения задачи. Особенность таких [языков программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) состоит в том, что задачи разбиваются на шаги и решаются шаг за шагом. Используя процедурный язык, программист определяет языковые конструкции для выполнения последовательности алгоритмических шагов.

1. **Перечислите группы операторов языка SQL.**

DDL – создание, удаление и изменение объектов БД или сервера СУБД

DML – операторы для работы с таблицами

TCL – для работы с транзакциями

DCL – для управления процессом авторизации

1. **Назовите основные операторы каждой группы.**

DDL – CREATE, ALTER, DROP

DML – SELECT, INSERT, DELETE, UPDATE

TCL – BEGIN TRAN, SAVE TRAN, COMMIT TRAN, ROLLBACK TRAN

DCL – GRANT, REVOKE, DANY

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

|  |  |
| --- | --- |
| Условное обозначение ограничения | Действие ограничения  целостности |
| **data type**  тип данных | Предотвращает появление в столбце значений, не соответствующих типу данных |
| **not null**  запрет значений **null** | Предотвращает появление в столбце значений null |
| **default**  знач. по умолчанию | Устанавливает значение в столбце по умолчанию при выполнении операции INSERT |
| **primary key**  первичный ключ | Предотвращает появление в столбце повторяющихся значений и пустого значения |
| **foreign key**  внешний ключ | Устанавливает связь между таблицей со столбцом, имеющим свойство **foreign key** (FK) и таблицей, имеющей столбец со свойством **primary key** (PK); предотвращает не согласованные операции между PK и FK |
| **unique**  уникальное значение | Не допускает пустые значения, не может быть использован для связи с **foreign key** |
| **check**  проверка значений | Предотвращает появление в столбце значения, не удовлетворяющего логическому условию |

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

FROM

WHERE

GROUP BY

HAVING

SELECT

DISTINCT

ORDER BY

TOP

INTO

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

INNER JOIN -к

LEFT OUTER JOIN

RIGHT OUTER JOIN

FULL OUTER JOIN -к

CROSS JOIN -к

1. **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

1. **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. Агрегатные функции часто используются в выражении GROUP BY инструкции SELECT.

\* **COUNT** - производит номера строк или не-NULL значения полей которые выбрал запрос.   
  
\* **SUM** - производит арифметическую сумму всех выбранных значений данного пол.   
  
\* **AVG** - производит усреднение всех выбранных значений данного пол.   
  
\* **MAX** - производит наибольшее из всех выбранных значений данного пол.   
  
\* **MIN** - производит наименьшее из всех выбранных значений данного пол.

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

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.

>=ALL формирует истинное значение в том случае, если значение стоящее слева больше или равно каждому значению в списке, указанном справа.

>=ANY формирует истинное значение в том случае, если значение стоящее слева, больше или равно хотя бы одному значению в списке, указанном справа.

BETWEEN позволяет указать диапазон значений

IS NULL сравнение с null

Like сравнение строк

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

1. **Перечислите этапы обработки SQL-запроса.**



1. **Поясните понятия «план запроса», «стоимость запроса».**

План запроса представляет собой алгоритм выполнения SQL-запроса

1. **Поясните понятия «индекс».**

**Индекс** – это объект базы данных, позволяющий **ускорить поиск** в определенной таблице.

1. **Перечислите известные вам типы индексов.**

Кластеризованные, некластеризованные, индекс покрытия, уникальные и фильтруемые некластеризованные индексы.

1. **В каких случаях и какого типа индексы создаются автоматически?**

Кластеризованные и уникальные не кластеризованные

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

операции добавления и изменения строк БД могут повлечь образование неиспользуемых фрагментов в области памяти индекса. Поэтому индексы таблиц, часто подвергающихся операциям изменения, становятся пористыми, или еще говорят – фрагментированными. Процесс образования неиспользуемых фрагментов памяти называют фрагментацией. Фрагментация индексов снижает эффект от их применения и в некоторых случаях может привести к значительным потерям производительности сервера БД.

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

Один, потом что они влияют на физический порядок строк в таблице

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

Для избавления от фрагментации индекса предусмотрены две специальные операции: реорганизация и перестройка индекса.

Реорганизация (REORGANIZE) выполняется значительно быстрее, но после этой операции фрагментация будет убрана только из листовых узлов индексного дерева (на самом нижнем уровне).

Операция перестройки (REBUILD) затрагивает все узлы дерева, поэтому после ее выполнения степень фрагментации равна нулю.

1. **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.

1. **В каких случаях к представлению применимы операторы INSERT, DELETE, UPDATE? Y**

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.

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

Чтобы операция вставки не могла осуществиться в том случае, когда информация не удовлетворяет условию, записанному в секции Where, то следует создавать представление с опцией WITH CHECK OPTION.

1. **Чем отличается оператор TRUNCATE от оператора DELETE?**

Оператор TRUNCATETABLE предназначен для удаления всех строк таблицы. Он похож на DELETE, но принцип его работы принципиально отличается. Выполнение DELETE осуществляется построчно с регистрацией каждой операции удаления в журнале транзакции. Оператор TRUNCATE table удаляет все строки одномоментно, а в журнале регистрируется только одна операция. Оператор TRUNCATE выполняется значительно быстрее.

1. **Поясните понятие «курсор». Перечислите типы курсоров известные вам.**

**Курсор** является программной конструкцией, которая служит для хранения результата запроса и дает возможность пользователю обрабатывать строки результирующего набора запись за записью

Локальные может применяться в рамках одного пакета и ресурсы, выделенные ему при объявлении, освобождаются сразу после завершения работы пакета.

Глобальные может быть объявлен, открыт и использован в разных пакетах. Выделенные ему при объявлении ресурсы освобождаются только после выполнения оператора DEALLOCATE или при завершении сеанса пользователя.

Динамические

Статические приводит к выгрузке результирующего набора во временную таблицу системной БД **TEMPDB**, и все дальнейшие операции осуществляются с этой таблицей, т. е. после открытия курсора все текущие изменения в исходных таблицах не будут отражаться в курсорном результирующем наборе.

1. **Поясните схему работы с курсором.**

курсор объявляется в операторе DECLARE, открывается с помощью оператора OPEN, с помощью оператора FETCH считывается одна или несколько строк результирующего набора. Результат каждого считывания проверяется с помощью системной функции @@FETCH\_STATUS. Затем курсор закрывается с помощью оператора CLOSE. Если курсор глобальный, то он должен быть освобожден с помощью оператора DEALLOCATE.

1. **Поясните назначение конструкции CURRENT OF при работе с курсором.**

Курсоры с установленным свойством FOR UPDATE помимо чтения данных из строк с помощью оператора FETCH, могут эти строки изменять или удалять с помощью операторов UPDATE и DELETE, если в секции WHERE эти операторы используют операцию CURRENT OF, для которой указывается имя курсора

1. **Поясните понятия «транзакция», «фиксация транзакции», «откат транзакции».**

**Транзакция** − это механизм базы данных, позволяющий таким образом объединять несколько операторов, изменяющих базу данных, чтобы при выполнении этой совокупности операторов они или все выполнились или все не выполнились.

Фиксация – подтверждение успешного завершения транзакции.

1. **Расшифруйте и поясните аббревиатуру ACID. Y**

Atomicity – атомарность, Consistency – согласованность, Isolation – изолированность, Durability – долговечность.

Основные свойства транзакции: *атомарность* (операторы изменения БД, включенные в транзакцию, либо выполнятся все, либо не выполнится ни один); *согласованность* (транзакция должна фиксировать новое согласованное состояние БД); *изолированность* (отсутствие взаимного влияния параллельных транзакций на результаты их выполнения); *долговечность* (изменения в БД, выполненные и зафиксированные транзакцией, могут быть отменены только с помощью новой транзакции).

1. **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.

Режим *неявной транзакции* может быть включен для текущего соединения с сервером БД с помощью специальной инструкции: SET IMPLICIT\_TRANSAC-TIONS ON. Обратное переключение осуществляется с использованием ключевого слова OFF вместо ON.

Транзакция начинается, если выполняется один из следующих операторов: CREATE, DROP; ALTER TABLE; INSERT, DELETE, UPDATE, SELECT, TRUNCATE TABLE; OPEN, FETCH; GRANT (выдача разрешений), REVOKE (запрещение разрешений).

Неявная транзакция продолжается до тех пор, пока не будет выполнен оператор фиксации (COMMIT) или оператор отката (ROLLBACK) транзакции.

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

1. **Перечислите известные вам уровни изолированности в порядке их усиления.**

|  |  |  |  |
| --- | --- | --- | --- |
| Уровень изоляции | Неподтвержденное  чтение | Неповторяющееся  чтение | Фантомное чтение |
| READ UNCOMMTITTED | + | + | + |
| READ COMMITED | – | + | + |
| REPEATABLE READ | – | – | + |
| SNAPSHOT | – | – | – |
| SERIALIZABLE | – | – | – |

1. **Поясните понятие «неподтвержденное чтение», «неповторяющееся чтение», «фантомное чтение». Y**
2. Чтение данных, добавленных или изменённых транзакцией, которая впоследствии не подтвердится (откатится).
3. Ситуация, когда при повторном чтении в рамках одной транзакции ранее прочитанные данные оказываются изменёнными.
4. Ситуация, когда при повторном чтении в рамках одной транзакции одна и та же выборка дает разные множества строк.



Рис. 13.46. Схема, демонстрирующая неподтвержденное чтение



Рис. 13.47. Схема, демонстрирующая неповторяющееся чтение



Рис. 13.48. Схема, демонстрирующая фантомное чтение

1. **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 БД, представляющий собой поименованный код T-SQL

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

Скалярная функция возвращает единственное значение и принимает любое количество параметров.

Встроенные табличные функции возвращают к точке вызова результирующий набор SELECT-запроса и по своим возможностям очень напоминают представления

Многооператорные табличные функции возвращают к точке вызова результирующий набор и допускают более сложную логику работы.

1. **Назовите отличия хранимых процедур от функций. Y**

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.

1. **Поясните понятия «DDL-триггер», «DML-триггер». Y**

Триггер – это особый вид хранимой процедуры, предназначенной для обработки событий в БД.

Различают два вида DDL-триггеров: триггеры уровня сервера (ALLSERVER), триггеры уровня БД (DATABASE). Триггеры уровня сервера могут обрабатывать события сервера СУБД (создание, изменение или удаление объектов сервера, подключение к серверу и пр.). Триггеры уровня базы данных предназначены для обработки событий, происходящих в рамках БД.

Каждый DML-триггер связан с таблицей или представлением и предназначен для обработки одного или нескольких событий, соответствующих трем операторам, изменяющим содержимое таблицы: INSERT, UPDATE и DELETE. MSS поддерживает два типа DML-триггеров: AFTER и INSTEAD OF.

*AFTER-триггеры*.Триггеры типа AFTER исполняются после выполнения оператора, вызвавшего соответствующее событие

*INSTEAD OF-триггеры*.Триггеры типа INSTEAD OF исполняются вместо оператора, вызвавшего соответствующее событие

1. **Поясните понятия «триггер уровня оператора», «триггер уровня строки», «событие, активизирующее триггер». Y**

Триггер первого уровня **Statement-level trigger** выполняется однократного после полного завершения одного из вышеуказанных операторов (**операторный триггер**) Триггер второго уровня **Row-level trigger** предназначен для многократного выполнения после каждого изменения одной из записей (строк) таблицы (**строчный триггер**).

1. Поясните смысл выражения «триггер является частью транзакции».
2. **Поясните понятия «XML», «XML-Schema», «W3C», «тег», «атрибут», «корневой тег», «XML-узел», «XML-элемент», «XML-документ», «элементная форма XML-документа», «атрибутная форма XML-документа»;**

XML (Extensible Markup Language) – расширяемый язык разметки

С помощью языка XML-Schema можно описать правила, которым должен подчиниться XML-документ. Файл, содержащий XML-Schema, обычно имеет расширение XSD (XML Schema definition). Большинство современных систем программирования предусматривают встроенные механизмы, позволяющие с помощью заданного XSD-файла проверять на корректность XML-документы.

атрибутивная модель сопоставления, каждый XML-атрибут преобразовывается в столбец таблицы

используется сопоставление на основе элементов, каждый элемент преобразовывается в столбец таблицы

1. Поясните понятия «XML-тип», «типизированные XML-данные», «коллекция XML-схем». Y