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

БД – это совокупность взаимосвязанных данных

1. Дайте определение понятию СУБД.

Програм. реализация **технологии** хранения, извлечения, обновления и обработки данных в бд

1. Назовите основные компоненты СУБД.

1)    **ядро**, которое отвечает за управление данными во внешней и оперативной памяти и журнализацию,

2)    **процессор языка базы данных**, обеспечивающий оптимизацию запросов на извлечение и изменение данных и создание, как правило, машинно-независимого исполняемого внутреннего кода,

3)    **подсистему поддержки времени исполнения**, которая интерпретирует программы манипуляции данными, создающие пользовательский интерфейс с СУБД

4)    а также **сервисные программы** (внешние утилиты), обеспечивающие ряд дополнительных возможностей по обслуживанию информационной системы.

1. Какие БД называют системными?

БД, которые поставляются вместе с СУБД называются системными

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

Отношение может быть представлено в виде двумерной таблицы

Реляционная база данных представляет собой набор взаимосвязанных таблиц

Все объекты разделяются на типы

Объекты одного и того же типа имеют свой набор атрибутов

Один из атрибутов однозначно идентифицирует объект в таблице – первичный ключ

* Структурный аспект  — данные в базе данных представляют собой набор отношений
* Аспект целостности — отношения (таблицы) отвечают определенным условиям целостности
* РМД поддерживает декларативные ограничения целостности уровня домена (типа данных), уровня отношения и уровня базы данных
* Аспект обработки — РМД поддерживает операторы манипулирования отношениями (реляционная алгебра, реляционное исчисление)

1. Что такое ***OLTP***?

OLTP (Online Transaction Processing) — обработка транзакций в реальном времени.

1. Что такое ***OLAP***?

OLAP (англ. online analytical processing, аналитическая обработка в реальном времени) — технология обработки информации, включающая составление и динамическую публикацию отчётов и документов.

1. Поясните понятие «архитектура клиент-сервер».

**«Клиент — сервер»** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *client–server*) — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами.

1. 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.

1. Поясните понятие «программный интерфейс СУБД».

Все СУБД имеют программные интерфейсы, которые позволяют подключаться к ним из сторонних приложений. Эти интерфейсы могут иметь вид программных библиотек различного уровня сложности и/или абстракции, а могут представлять собой драйверы для распространенных систем взаимодействия с СУБД, как например JDBC (Java Database Connectivity) или ODBC (Open Database Connectivity).

1. Поясните понятие «интерфейс командной строки».

**Интерфейс командной строки** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *Command line interface, CLI*) — разновидность [текстового интерфейса](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) (CUI) 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.

**SQL** ([ˈɛsˈkjuˈɛl](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B6%D0%B4%D1%83%D0%BD%D0%B0%D1%80%D0%BE%D0%B4%D0%BD%D1%8B%D0%B9_%D1%84%D0%BE%D0%BD%D0%B5%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BB%D1%84%D0%B0%D0%B2%D0%B8%D1%82); [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *structured query language* — «язык структурированных запросов») — [декларативный](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BA%D0%BB%D0%B0%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B5_%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%D0%B5) [язык 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), применяемый для создания, модификации и управления данными в [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](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85).

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

В декларативном языке пишешь, что нужно сделать, но не описываешь как.

При декларативном подходе выражается логика вычисления без отсутствия описания потока управления, тогда как в императивной парадигме программирования используются утверждения, изменяющие состояние программы.

SQL, Prolog.

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

В процедурном языке нужно описывать каждый шаг программы

 PHP

 [GNU bc](https://ru.wikipedia.org/wiki/Bc)

 [Euphoria](https://ru.wikipedia.org/wiki/Euphoria_(%D1%8F%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))

 [Limbo](https://ru.wikipedia.org/wiki/Limbo)

 [Lua](https://ru.wikipedia.org/wiki/Lua)

 [Maple](https://ru.wikipedia.org/wiki/Maple)

 [MATLAB](https://ru.wikipedia.org/wiki/MATLAB)

1. Перечислите группы операторов языка SQL.

**DDL** - Data Definition Language - язык определения данных

**DML** - Data Manipulation Language - язык манипулирования данными

**TCL** - Transaction Control Language - язык управления транзакциями

**DCL** - Data Control Language - язык управления данными

1. Назовите основные операторы каждой группы.

DDL включает операторы:

* 1. **CREATE**
  2. **ALTER**
  3. **DROP**

DML включает операторы:

* 1. **SELECT**
  2. **INSERT**
  3. **DELETE**
  4. **UPDATE**

TCL SQL включает операторы:

* 1. **BEGIN TRAN**
  2. **SAVE TRAN**
  3. **COMMIT TRAN**
  4. **ROLLBACK TRAN**

DCL включает в себя операторы:

* 1. **GRANT**
  2. **REVOKE**

**DENY**

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

**Ограничения целостности** - совокупность правил, позволяющих обеспечить в любой момент времени правильность данных (непротиворечивость, удовлетворяемость, адекватность существующим знаниям о реальном мире).

PRIMARY KEY, UNIQUE,NULL,NOT NULL,тип данных,foreign key, check,default

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.

внутреннее объединение таблиц (**INNER JOIN**), внешнее объединение таблиц (**LEFT OUTER JOIN, RIGHT JOIN, FULL JOIN**) и перекрестное объединение таблиц (**CROSS JOIN**).

Коммутативны – FULL OUTER JOIN, CROSS JOIN, INNER JOIN

1. Поясните понятия «подзапрос», «некоррелируемый подзапрос» и «коррелируемый подзапрос».

**Подзапрос** – это SELECT-запрос, который выполняется в рамках другого запроса.

*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.

Агрегатная функция – функция вычисляющая значение для набора строк.

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

1. 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.

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

1. Перечислите этапы обработки SQL-запроса.

 СУБД сначала выполняет синтаксический анализ инструкции SQL. Он разбивает инструкцию на отдельные слова, называемые маркерами, гарантирует, что инструкция имеет допустимую команду и допустимые предложения и т. д. На этом шаге можно обнаружить синтаксические ошибки и орфографических ошибок.

 СУБД проверяет инструкцию. Он проверяет выполнение инструкции в системном каталоге. Существуют ли все таблицы, с именем в инструкции в базе данных? Все столбцы существуют и однозначны имена столбцов? Есть ли у пользователя недостаточно прав для выполнения инструкции? На этом шаге можно обнаружить некоторые семантические ошибки.

 СУБД формирует план доступа для инструкции. План доступа является двоичным представлением действия, необходимые для выполнения инструкции. он эквивалентен СУБД исполняемого кода.

 СУБД оптимизирует плана доступа. Он исследует различные способы выполнения плана доступа. Можно использовать индекс для ускорения поиска? Сначала необходимо применить условие поиска в таблице А необходимо СУБД и затем присоедините ее к таблице B, или необходимо его начинаются с соединения и впоследствии использовать условие поиска? Можно последовательного поиска по таблице избегать или уменьшен до подмножества таблицы? После изучения альтернативные решения, СУБД выбирает один из них.

 СУБД выполняет инструкцию, выполнение плана доступа.

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

**План выполне́ния запро́са** — последовательность операций, необходимых для получения результата [SQL](https://ru.wikipedia.org/wiki/SQL)-запроса в [реляционной СУБД](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94).

**Стоимость запроса** – количество ресурсов потраченное на выполнение запроса.

1. Поясните понятия «индекс».

**Индекс** – это объект базы данных, позволяющий **ускорить поиск** в определенной таблице, так как при этом данные организуются в виде сбалансированного бинарного дерева поиска.

1. Перечислите известные вам типы индексов.

кластеризованные, некластеризованные, уникальные, неуникальные, составные, *индекс покрытия, фильтруемый индекс*

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

Обычно *кластеризованные* индексы создаются автоматически при создании таблицы при наличии первичного ключа (ограничение PRIMARY KEY).

1. Поясните понятие «фрагментация индекса».

Операции добавления и изменения строк базы данных могут повлечь образование неиспользуемых фрагментов в области памяти индекса. Процесс образования неиспользуемых фрагментов памяти называется *фрагментацией*.

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

Кластеризованные индексы физически упорядочены в соответствии со значениями индексируемых столбцов. В таблице может быть только один кластеризованный индекс.

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

*Реорганизация* (REORGANIZE) выполняется быстро, но после нее фрагментация будет убрана только 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.

Представление создается с помощью оператора CREATE

1. 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.

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?

Delete - это DML, Truncate - это 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

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.

Поскольку TRUNCATE TABLE не логируется, то и не может активировать триггер.

Откат (rollback) после TRUNCATE невозможен

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

**Курсор – механизм, позволяющий обрабатывать отдельные строки, полученные в результате select-запроса.** Курсоры бывают *локальные* и *глобальные* (по умолчанию), *статические* и *динамические* (по умолчанию).

1. Поясните схему работы с курсором.

Курсор объявляется в операторе DECLARE.

Курсор открывается с помощью оператора OPEN.

С помощью оператора FETCH считывается одна или несколько строк результирующего набора, связанного с курсором SELECT-оператора, и обрабатывается нужным образом. Результат каждого считывания проверяется с помощью системной функции @@FETCH\_STATUS.

Курсор закрывается оператором CLOSE.

Если курсор глобальный, то он должен быть освобожден с использованием оператора DEALLOCATE.

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

Для изменения строки соответствующей текущей позиции 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 транзакции – подтверждение выполнения операций в транзакции;

Откат транзакции – подтверждение не выполнения операций в транзакции;

1. Расшифруйте и поясните аббревиатуру ACID.

Atomicity - Атомарность

Consistency - Согласованность

Isolation - Изолированность

Durability – Долговечность

*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».

В режиме автоматической фиксации каждой операции базы данных является транзакцией, которая фиксируется при выполнении.

В режиме явной транзакции начало и конец транзакции явно указан в коде.

Режим неявной транзакции - если транзакцию явно не зафиксировать, то все изменения, выполненные в ней, откатываются при отключении пользователя.

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

READ UNCOMMITTED

READ COMMITTED

REPEATABLE READ

SERIALIZABLE

SNAPSHOT

1. Поясните понятие «неподтвержденное чтение», «неповторяющееся чтение», «фантомное чтение».

*Неподтвержденное чтение.* До момента t1 транзакцией B выполняются два оператора: INSERT и UPDATE. Эти операторы изменяют таблицы БД, но до момента времени t2 не фиксируют и не откатывают эти изменения. После момента t1 транзакция A считывает содержимое таблиц, измененных транзакцией B и «видит» измененные или добавленные строки. При этом изменения остаются до момента t2 в неподтвержденном состоянии, т. е. могут быть как зафиксированными, так и отмененными.

*Неповторяющееся чтение.* Одна транзакция читает данные несколько раз, а другая изменяет те же данные между двумя операциями чтения в первом процессе. По этой причине данные, прочитанные в различных операциях, будут разными.

*Фантомное чтение.* Две последовательные операции чтения могут получать различные значения, т. к. дополнительные строки, называемые фантом­ными, могут добавляться другими транзакциями.

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 поименованный код на языке Transact-SQL.

**Параметрическая форма передачи параметров** - это такая передача параметров, в которой указывается какому параметру какой соответствует, а **позиционная** - это такая форма, в которой параметры указываются в том порядке, который был указан при объявлении функции или процедуры.

**Системная хранимая процедура** – процедура поставляемые разработчиками СУБД.

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

**Скалярная функция** – функция возвращающая одно значение;

**Встроенная табличная функция** – функция, встраиваемая в запрос, которая возвращает переменную типа table.

**Многооператорная хранимая функция** –функция, возвращающая набор данных.

1. Назовите отличия хранимых процедур от функций.

Вызов процедуры осуществляется оператором EXECUTE (EXEC). В хранимых процедурах допускается применение основных DDL, DML и TCL-операторов, конструкций TRY/CATCH, курсоров, временных таблиц.

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».

DML-триггеры – триггеры создаются для таблицы или представления.

DDL-триггеры – триггеры уровня сервера или базы данных.

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

Триггер уровня оператора вызывается не для каждой модифицируемой записи, а один раз на изменение таблицы.

Триггер уровня строки вызывается для каждой модифицируемой записи.

Событие, активизирующее триггер – событие, приводящее к срабатыванию триггера.

1. Поясните смысл выражения «триггер является частью транзакции».

Ошибка в триггере откатывает операцию, изменения таблиц в триггере становятся частью транзакции.

1. 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** ( *e****X****tensible* ***M****arkup* ***L****anguage*) — расширяемый [язык разметки](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B8).

**Консо́рциум Всеми́рной паути́ны** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *World Wide Web Consortium,* ***W3C***) — организация, разрабатывающая и внедряющая технологические стандарты для [Всемирной паутины](https://ru.wikipedia.org/wiki/%D0%92%D1%81%D0%B5%D0%BC%D0%B8%D1%80%D0%BD%D0%B0%D1%8F_%D0%9F%D0%B0%D1%83%D1%82%D0%B8%D0%BD%D0%B0).

**XML Schema** — язык описания структуры **XML**-документа.

**Тег** –элемент языка разметки гипертекста.

**атрибут** — пара имя-значение, добавляемые в открывающий тег после названия элемента.

**XML-узел** – любая часть XML документа.

**XML элемент** — это все от (и включая) начального тега элемента до (и включая) конечного тега элемента.

**XML-документ** представляет собой обычный текстовый файл, в котором при помощи специальных маркеров создаются элементы данных, последовательность и вложенность которых определяет структуру документа и его содержание.

**Корневой тег** –тег с которого начинается документ.

**атрибутная форма XML-документа** - каждый XML-атрибут преобразовывается в столбец таблицы

**элементная форма XML-документа** - каждый элемент преобразовывается в столбец таблицы

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

**XML-тип** – тип, который сообщает бд, что это XML.

**Типизированные XML-данные** – данные, которым соответствует XML-схема.

Коллекция XML-схем хранит импортированные XML-схемы и используется для решения следующих задач:

* проверка экземпляров XML;
* типизация XML-данных, хранимых в базе данных.