Учреждение образования
«Гомельский государственный университет имени Франциска Скорины»

Факультет физики и информационных технологий
Кафедра автоматизированных систем обработки информации

СОГЛАСОВАНО
Заведующий кафедрой
автоматизированных систем
обработки информации
А.В.Воруев
__ 18.09 2023

СОГЛАСОВАНО
Декан
факультета физики
и информационных технологий
Д.Л.Коваленко
_____ 2023

ЭЛЕКТРОННЫЙ УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

МЕТОДЫ ОБРАБОТКИ БОЛЬШИХ МАССИВОВ ДАННЫХ

для специальности 1-45 80 01 Системы и сети инфокоммуникаций
второй ступени высшего образования (магистратура)

Составитель: старший преподаватель кафедры АСОИ Леванцов В.Н.
старший преподаватель кафедры АСОИ Пугачева Е.Е.
ассистент кафедры АСОИ Рафалова Е.В.

Рассмотрено и утверждено
на заседании кафедры АСОИ
__18 апреля___ 2023 г., протокол № _9_

Рассмотрено и утверждено
на заседании научно-методического
совета университета
__29 апреля___ 2023 г., протокол № _8_

Гомель 2020

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Электронный учебно-методический комплекс (ЭУМК) по дисциплине «Методы обработки больших массивов данных» представляет собой комплекс систематизированных учебных, методических и вспомогательных материалов, предназначенных для использования в образовательном процессе специальности 1-45 80 01 Системы и сети инфокоммуникаций.

ЭУМК разработан в соответствии со следующими нормативными документами:

1. Положением об учебно-методическом комплексе на уровне высшего образования, утвержденном постановлением Министерства образования Республики Беларусь от 08.11.2022 № 427.

2. Учебных планов УВО специальности высшего образования второй ступени (магистратура) 1-45 80 01 Системы и сети инфокоммуникаций регистрационный № I 45-2-01/Д-19 от 09.04.2019 г. и № I 45-2-01/З-19 от 09.04.2019 г.

3. Учебной программой по учебной дисциплине «Методы обработки больших массивов данных» для специальности 1-45 80 01 Системы и сети инфокоммуникаций, утвержденной 20.05.2020, регистрационный номер УД-31-2020-131/уч.

Цель создания ЭУМК по дисциплине «Методы обработки больших массивов данных» является овладение основами обработки больших массивов данных. Обучающиеся должны сформировать целостное представление о современных проблемах анализа и обработки больших данных, помочь овладеть опытом разработки и анализа концептуальных и теоретических моделей прикладных задач анализа больших данных с применением моделей Data Mining, разрабатывать маркетинговую стратегию организаций.

ЭУМК направлен на оказание помощи обучающимся в овладении методами количественного анализа и моделирования, теоретического и экспериментального исследования; основными методами, способами и средствами получения, хранения, переработки информации, навыками работы с компьютером как средством управления информацией; методами и программными средствами обработки деловой информации, способностью взаимодействовать со службами информационных технологий и эффективно использовать корпоративные информационные системы.

ЭУМК способствует успешному осуществлению учебной деятельности, дает возможность планировать и осуществлять самостоятельную работу обучающихся, обеспечивает рациональное распределение учебного времени по темам учебной дисциплины и совершенствование методики проведения занятий.

В структурном отношении ЭУМК включает четыре раздела: теоретический, лабораторный практикум, тесты, раздел контроля знаний и вспомогательный.

Теоретический раздел содержит лекционный материал включающий в себя шестнадцать тем:

Тема 1 Введение в большие данные.

Тема 2 Процесс анализа.

Тема 3 Корреляция и регрессионный анализ.

Тема 4 Задачи классификации и кластеризации.

Тема 5 Технологии KDD и Data Mining.

Тема 6 Парадигма Map Reduce.

Тема 7 Программное обеспечение в области анализа данных.

Тема 8 Подготовка данных.

Тема 9 Проблема переобучения.

Тема 10 Основные понятия теории нейронных сетей.

Тема 11 Определение дерева решений.

Причины популярности и условия применимости. Структура дерева решений. Выбор атрибута разбиения в узле. Алгоритм ID3, критерий выбора атрибута разбиения ID3, пример работы алгоритма. Проблема переобучения, Неизвестные значения атрибутов, алгоритм C4.5.

Лабораторный раздел включает в себя темы лабораторных занятий. В каждой теме содержится краткое изложение теоретического материала и задания для выполнения по вариантам.

Вспомогательный раздел содержит необходимые элементы учебно-программной документации.

Все разделы ЭУМК в полной мере соответствуют содержанию и объему образовательного стандарта.

Общее количество часов – 126. (4 зачетных единицы)

Дневная форма обучения: аудиторное количество часов – 46; из них: лекционных занятий – 22 (в том числе УСР – 4), практических занятий – 12(в том числе УСР – 10), лабораторных работ – 12 (в том числе УСР – 4) .

Форма отчётности – экзамен.

Заочная форма обучения: аудиторное количество часов – 10; из них: лекционных занятий – 6, практических занятий – 2, лабораторных работ – 2.

Форма отчётности – экзамен.

ТЕКСТ ЛЕКЦИЙ

Содержание

Введение

Объем данных, генерируемый и собираемый современными научно-исследовательским центрами, финансовыми институтами, социальными сетями, уже привычно измеряется петабайтами. Так в дата-центрах Facebook хранится уже более 15 млрд. изображений, нью-йоркская фондовая биржа NYSE создает и реплицирует ежедневно около 1 Тб данных, Большой адронный коллайдер получает около 1 Пб данных в секунду.
Таким образом, в современном мире возникла проблема больших данных или Big Data. Мировые лидеры в сфере ИТ и бизнеса заняты поиском оптимального решения для управления огромным количеством постоянно прибывающей информации и ее анализа. Они ищут пути извлечения выгоды из данных находящихся в их распоряжении.

Тема больших данных интересна как с практической, так и с теоретической точек зрения. Сами технологии находятся в состоянии непрерывного развития, что позволяет как в режиме реального времени наблюдать за процессом их внедрения и совершенствования, так и непосредственно участвовать в создании новых технологий обработки больших массивов данных.

Также хочется заметить, что расширение познаний и навыков в сфере Big Data является особенно актуальным для студентов специальности бизнес-информатика.

1. Большие данные

Описывая специфику больших данных, первым делом упоминают 3V: "volume, variety and velocity" или объем, разнообразие и скорость.

Объем подразумевает не только большое количество хранимой информации, но и ее дополнение, рост, изменение с течением времени.

Разнообразие типов и источников информации всегда было большой проблемой, когда появлялась необходимость свести их в один массив данных. Сегодня это разнообразие только увеличивается.

Скорость оценивается как при создании информации, так и при ее обработке.

Традиционным методом работы с массивами информации являются реляционные базы данных. Однако работа с реляционной базой данных на сотни терабайт - это еще не Big Data, а, например, "обычная" highload-БД. Разница, в данном случае, заключается в архитектуре БД и логике взаимодействия СУБД с хранящейся информацией.

В реляционных БД информация распределена дисперсионно, т.е. имеет место изначально заданная четкая структура, изменение которой в уже работающей базе связано с множеством проблем. Таким образом, в силу своей архитектуры, реляционные БД лучше всего подходят для коротких быстрых запросов, идущих однотипным потоком. Сложный же запрос либо потребует перестройки структуры БД, либо, в угоду быстродействию, увеличения вычислительных мощностей. Это указывает на еще одну проблему традиционных баз данных, а именно на сложность их масштабируемости.

Таким образом, для работы со сложными гибкими запросами необходима среда, позволяющая хранить и обрабатывать неструктурированные данные, поддающаяся масштабированию и допускающая применения распределенных вычислений, где для обработки данных используется не одна высокопроизводительная машина, а целая группа таких машин, объединенных в кластер.

2. Техники анализа больших данных

На данный момент существует и разрабатывается множество техник анализа больших кластеров информации. Далее будут приведены некоторые из них.

Слияние и интеграция данных (Data fusion and data integration). Набор техник, которые сводят вместе и анализируют информацию из различных источников, с целью получения более достоверной и, в перспективе, полезной информации, чем при использовании единственного источника. Для этого может быть использована **обработка цифровых сигналов** (Signal processing). Например, данные СМИ, проанализированные с **помощью обработки естественного языка** (natural language processing) и сопоставленные с данными о продажах, могут выявить механизм воздействия рекламных компаний и другой информации на поведение покупателей.
Интеллектуальный анализ данных (Data mining). Набор техник извлечения потенциально полезной информации из больших массивов данных путем комбинации различных методов, от **статистики** до **машинного обучения** (machine learning) и **управления базами данных** (database management). Они включают в себя **ассоциативное обучение** (association rule learning), **кластерный анализ** (cluster analysis), **классификацию** и **регрессию**.
Генетические алгоритмы (Genetic algorithms). Техника, используемая для оптимизации и основанная на принципе естественной эволюции: "выживание наиболее приспособленного". Здесь потенциальные решения внесены в код подобно хромосомам и могут составлять комбинации и мутировать. Также часто описываются как тип **эволюционных алгоритмов**, хорошо подходящих для решения нелинейных задач. Примером может являться улучшение рабочего графика или оптимизация инвестиционного портфеля.
Нейронные сети (Neural networks). Вычислительные модели, вдохновленные нервной системой человека и животных. Они хорошо подходят для нахождения сложных образов, и могут быть использованы для распознавания и оптимизации. Могут включать в себя, так называемое, **обучение с учителем** (supervised learning), или **обучение без учителя** (unsupervised learning)
Обработка потоков (Stream processing). Технологии обработки большого количества потоков данных о событиях в реальном времени.
Также широко используется метод **визуализации** и другие.

3. Технологии анализа больших данных

Технологии анализа, в силу перспективности развития этого направления и большой коммерческой отдачи, также широко представлены, и их число продолжает расти. Ниже приведен список и описание наиболее заметных из них.

Business intelligence (BI). Прикладное программное обеспечение, разработанное для сбора, анализа и представления данных. Инструменты IB часто используются для создания стандартных отчетов или для отображения информации в реальном времени на панелях управления.

Cassandra. Бесплатная СУБД с открытым кодом, предназначенная для обращения с большим количеством данных на базе распределенной системы. Изначально разработана в Facebook, сейчас числится как проект the Apache Software foundation
Extract, transform, and load (ETL). Программные инструменты для извлечения данных их внешних источников, адоптации их под стандарт системы и загрузки в базу данных.
Google File System. Фирменная распределенная фаиловая система Гугла.
Hadoop. Бесплатная программная среда для обработки огромных массивов данных и решения определенных типов задач на основе распределенных систем. Разработка вдохновлена Google's MapReduce и Google File System. Изначально разработана в Yahoo!, сейчас числится как проект the Apache Software foundation.

4. Рынок СУБД. Игроки и тренды

В феврале 2012 года исследовательская и консалтинговая компания Gartner представила свой аналитический отчет для хранилищ данных. В рамках этого отчета хранилища данных определены, как СУБД, которая управляет и поддерживает одну или несколько логических баз данных в хранилище. Кроме этого, СУБД хранилища данных должна поддерживать реляционную модель данных, а также иметь возможность предоставить доступ к данным через программные интерфейсы для того, чтобы сторонние аналитические приложения могли воспользоваться данными, находящимися в хранилище данных. В дополнение к этому, СУБД хранилища данных должна иметь механизмы, изолирующую различные типы нагрузок друг от друга, а также управлять различными параметрами доступа пользователей в рамках одного экземпляра данных.

По результатам анализа, проведенного в этом отчете, Gardner составил так называемый Magic Quadrant, где разместил компании соответственно полям этого квадрата. большое анализ массив информация

Очевидно, что кампании, находящиеся в правом верхнем углу, представили самые успешные решения, что гарантировало им лидерство в данной отрасли. Естественно, борьба между этими компаниями идет и в сфере анализа больших данных. Тут можно выделить два основных подхода:

Адаптационный. Разработчик оптимизирует систему для работы с большими данными, при этом не меняя логику и архитектуру существенно, а, в основном, дополняя и дорабатывая готовый продукт.
Революционная. Создание качественно нового продукта, использующего принципиально другую логику, например, NoSQL, и разработанного специально для анализа массивов неструктурированной информации.
Сложно сказать, какой из этих подходов правильней. Первое решение будет востребовано, т.к. реляционные БД используются сейчас и будут использоваться в обозримом будущем, а потому на технологию анализа больших данных в их рамках будет спрос. Второй подход также успешно применяется уже несколько лет и приносит коммерческую выгоду. Также это острие аналитической мысли, которое привлекает многих специалистов.

Еще хочется отметить, что тенденция в области хранилищ данных относительно архитектуры такова, что в будущем останутся только решения, основанные на MPP (Massive Parallel Processing) архитектуре, так как именно они позволяют обрабатывать огромные объемы информации на стандартном аппаратном обеспечении.

Массово-параллельная архитектура (Massive Parallel Processing, MPP) - это класс параллельных вычислительных систем, состоящих из множества узлов, где каждый узел представляет собой автономную, независимую от других единицу. Если применить это определение к области хранилищ данных, то лучше всего его смысл будет отражать термин "**распределённые базы данных**". Каждый узел в распределенной базе данных представляет собой полноценную СУБД, работающую независимо от других. Сама же распределенная база данных - это совокупность независимых, автономных узлов, связанных коммуникационной сетью. Все данные в такой сети распределяются между

узлами равномерно, т.е. каждый узел хранит свою, уникальную данных, логически, тем не менее, представляя единую базу данных.

Заключение

Большие данные, появившиеся как следствие движения общества по информационному пути развития, уже стали частью нашей ежедневной жизни. Почти каждый человек ежедневно генерирует информацию, которая обрабатывается и записывается на различного рода носители. Неудивительно, что правительство и бизнес, в их извечной гонке за эффективностью, крайне заинтересованы в анализе этой информации, что в свою очередь, подогревает интерес разработчиков к данной сфере.

Но необходимо помнить, что попав в круг пристального внимания прессы и инвесторов, "Big data" не стали чем-то качественно новым. Разработанные технологии анализа носят, скорее, количественный характер, и их развитие обусловлено, в первую очередь, появлением нового оборудования, обладающего большой вычислительной мощностью и возможностью записи огромных объемов информации.

Также, несмотря на наличие большого количества идей по использованию технологий "Big data" в социальной среде, их первоначальной целью было и остается извлечение прибыли.

Нейронные сети

Нейронные сети - класс аналитических методов, построенных на (гипотетических) принципах обучения мыслящим существ и функционированию мозга, которые позволяют прогнозировать значения некоторых сменных в новых наблюдениях на основе результатов других наблюдений (для этих же или других сменных) после прохождения этапа так называемого обучения на имеющихся данных.

1. Основные понятия о нейронных сетях

Наиболее часто нейронные сети используются для решения следующих задач:

- классификация образов - указание на принадлежность входного образа, представленного вектором признаков, одному или нескольким предварительно определенным классам;
- кластеризация - классификация образов при отсутствии учебной выборки с метками классов;
- прогнозирование - предусмотрение значения y(tn+1) при известной последовательности y(t1), y(t2) ... y(tn);
- оптимизация - обнаружение решения, которое удовлетворяет систему ограничений и максимизирует или минимизирует целевую функцию. Память, которая адресуется по смыслу (ассоциативная память) - память, доступная при указании заданного содержания;
- управление - расчет такого входного влияния на систему, за который система работает по желательной траектории.

Структурной основой нейронной сети является формальный нейрон. Нейронные сети возникли из попыток воссоздать способность биологических систем учиться, моделируя низкокорневую структуру мозга. Для этого в основу нейросетевой модели ложится элемент, который имитирует в первом приближении свойства биологического нейрона - формальный нейрон(далее просто нейрон). В организме человека нейроны это особые клетки, способны распространять электрохимические сигналы.

Нейрон имеет разветвленную структуру для введения информации (дендриты), ядро и выход, который разветвляется (аксон). Будучи соединенными определенным образом, нейроны образовывают нейронную сеть. Каждый нейрон характеризуется определенным текущим состоянием и имеет группу синапсов - однонаправленных входных связей, соединенных с выходами других нейронов, а также имеет аксон - исходная связь данного нейрона, за которым сигнал (нарушение или торможение) поступает на синапсы следующих нейронов (рис. 8.1).

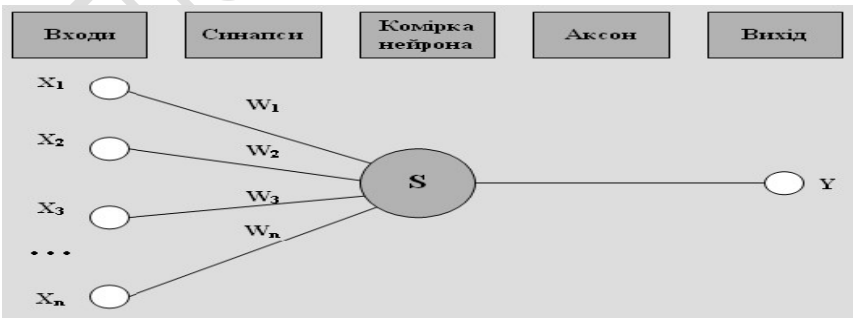


Рис. 8.1. Структура формального нейрона.

Каждый синапс характеризуется величиной синапсичной связи или его весом wi, что по физическому содержанию эквивалентная электрической проводимости.

Текущее состояние (уровень активации) нейрона определяется, если взвешенная сумма его входов:

$$S=\sum_{i=1}^{n} x_i * w_i \quad (1)$$

где множество сигналов, обозначенных x1, x2,..., xn, поступает на вход нейрона, каждый сигнал увеличивается на соответствующий вес w1, w2,...,wn,и формирует уровень его активации - S. Выход нейрона есть функция уровня его активации:

Y=f(S) (2)

При функционировании нейронных сетей выполняется принцип параллельной обработки сигналов. Он достигается путем объединения большого числа нейронов в так называемые пласты и соединения определенным образом нейронов разных пластов, а также, в некоторых конфигурациях, и нейронов одного пласта между собой, причем обработка взаимодействия всех нейронов ведется послойно.

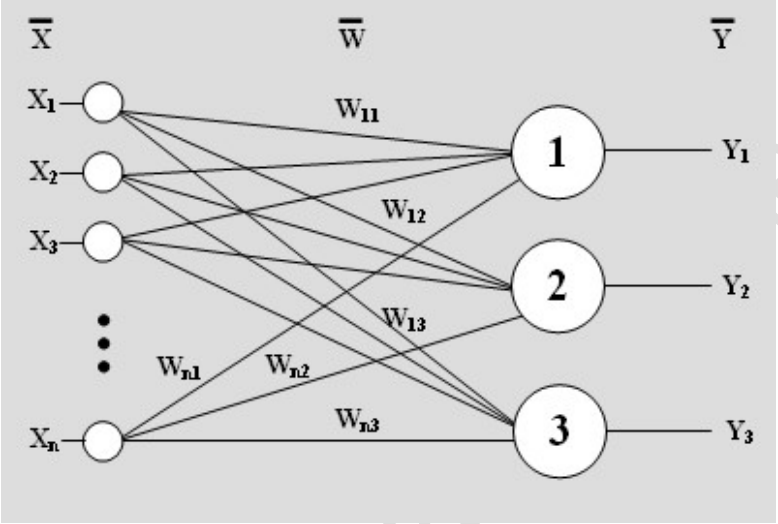


Рис. 8.2. Архитектура нейронной сети с n нейронами во входном и тремя нейронами в исходном пласте (однослойный персептрон).

В качестве примера простейшей нейронной сети, рассмотрим однослойный персептрон с n нейронами во входном и тремя нейронами в исходном пласте (рис. 8.2). Когда на n входов поступают какие-то сигналы, они проходят по синапсам на 3 исходные нейрона. Эта система образовывает единый пласт нейронной сети и выдает три исходных сигнала:

Очевидно, что все весовые коэффициенты синапсов одного пласта нейронов можно свести в матрицу wj, каждый элемент которой wij задает величину синапсичной связи i-го нейрона входного и j-го нейрона исходного пласта(3).

$$Y_j=f\left[\sum_{i=1}^{n} x_i * w_{ij}\right] \quad (3)$$

Таким образом, процесс, который происходит в нейронной сети, может быть записан в матричной форме:

$$Y_j=f\left(xw_j\right), \quad (4)$$

где x и y - соответственно входной и исходный векторы, f(v) - активационная функция, которая применяется поэлементно к компонентам вектора v.

Выбор структуры нейронной сети осуществляется согласно особенностям и сложности задачи. Для решения некоторых отдельных типов задач уже существуют оптимальные конфигурации. Если же задача не может быть сведена ни к одному из известных типов, разработчику приходится решать сложную проблему синтеза новой конфигурации.

Возможная такая классификация существующих нейросетей:

По типу входной информации:

- сети, которые анализируют двоичную информацию;
- сети, которые оперируют с действительными числами.

По методу обучения:

- сети, которые необходимо научить перед их применением;
- сети, которые не нуждаются в предыдущем обучении, способны обучаться самостоятельно в процессе работы.

По характеру распространения информации:

- однонаправленные, в которых информация распространяется только в одном направлении от одного пласта к другому;
- рекурентные сети, в которых исходный сигнал элемента может снова поступать на этот элемент и другие элементы сети этого или предыдущего пласта как входной сигнал.

По способу преобразования входной информации:

- автоассоциативные;
- гетероассоциативные.

Развивая дальше вопрос о возможной классификации нейронных сетей, важно отметить существования бинарных и аналоговых сетей. Первые оперируют с двоичными сигналами, и выход каждого нейрона может принимать только два значения: логический нуль ("приостановленное" состояние) и логическая единица ("возбужденное" состояние). Еще одна классификация разделяет нейронные сети на синхронные и асинхронные. В первом случае в каждый момент времени свое состояние изменяет лишь один нейрон. Во второму - состояние изменяется сразу у целой группы нейронов, как правило, во всем пласте.

Сети также можно классифицировать по количеству пластов. На рис. 8.3 представлен двухслойный персептрон, полученный из персептрона на рис. 8.2 путем добавления второго пласта, который состоит из двух нейронов.

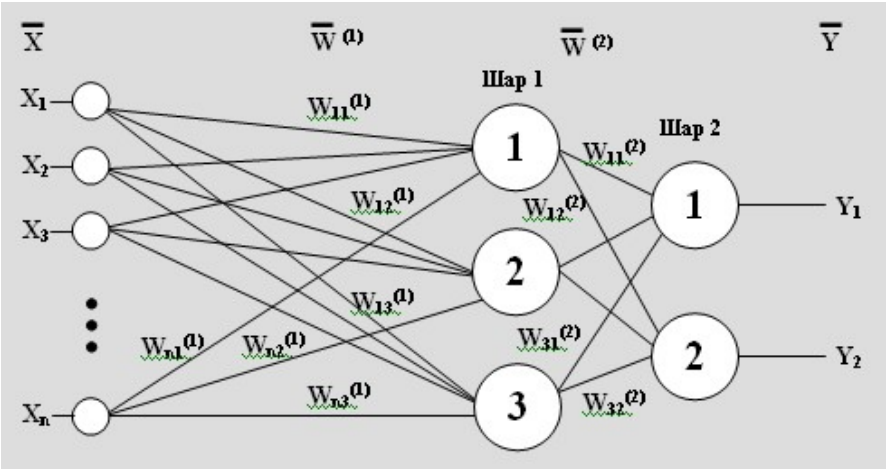


Рис. 8.3. Архитектура нейронной сети с однонаправленным распространением сигнала – двухслойный персептрон.

Если рассматривать работу нейронных сетей, которые решают задачу классификации образов, то вообще их работа сводится к классификации (обобщения) входных сигналов, которые принадлежат n-мерному гиперпространству, по некоторому числу классов. С математической точки зрения это происходит путем разбивки гиперпространства гиперплоскостями (запись для случая однослойного персептрона)

$$\sum_{i=1}^{n} x_i * w_{ik} = T_k \quad , (5),$$

где *k=1...m* – номер класса.

Каждая полученная область является областью определения отдельного класса. Число таких классов для одной нейронной сети персептронного типа не превышает 2m, где m - число выходов сети. Однако не все из них могут быть распределены данной нейронной сетью.

Технологии Data Mining и KDD

Я очень заинтересован в процессах. Я хочу знать хорошие способы сделать что-то, даже лучший способ сделать что-либо, если это возможно. Даже если у вас нет навыков или глубокого понимания, процесс может проделать долгий путь. Он может проложить путь, за которым последуют навыки и глубокое понимание. По крайней мере, я использовал это, чтобы вести большую часть моей работы.

Я думаю, что это полезно для изучения интеллектуального анализа данных, поскольку он представлен как процесс открытия данных. В этом посте вы изучите авторитетные определения «интеллектуального анализа данных» из учебников и статей. Поскольку интеллектуальный анализ данных является процессом, определение будет включать в себя ряд интерпретаций процесса.

«Интеллектуальный анализ данных - это извлечение неявной, ранее неизвестной и потенциально полезной информации из данных. Идея состоит в том, чтобы создавать компьютерные программы, которые автоматически просматривают базы данных в поисках закономерностей или закономерностей. Сильные шаблоны, если они будут найдены, скорее всего, будут обобщать, чтобы делать точные прогнозы на будущие данные. ... Машинное обучение обеспечивает техническую основу для интеллектуального анализа данных. Он используется для извлечения информации из необработанных данных в базах данных... »

«Интеллектуальный анализ данных определяется как процесс обнаружения закономерностей в данных. Процесс должен быть автоматическим или (чаще) полуавтоматическим. Обнаруженные закономерности должны быть значимыми, поскольку они приводят к некоторому преимуществу, обычно экономическому. Данные неизменно присутствуют в значительных количествах ».

Я прочитал эту книгу в начале моего вступления в поле, и это определение интеллектуального анализа данных и его связи с машинным обучением застряло у меня. Когда я применяю методы машинного обучения, я применяю процесс, который похож на процесс интеллектуального анализа данных, за исключением того, что я не пытаюсь обнаружить шаблоны как таковые, а пытаюсь найти «достаточно хорошее» решение хорошо определенной проблемы.

Интеллектуальный анализ данных: концепции и методы

«Интеллектуальный анализ данных, также широко известный как обнаружение знаний из данных (KDD), представляет собой автоматизированное или удобное извлечение шаблонов, представляющих знания, неявно хранящиеся или захваченные в больших базах данных, хранилищах данных, Интернете, других массивных хранилищах информации или потоках данных».

Это немного другое определение KDD, которое я считаю стандартным в данной области. Я считаю, что предпочтительным определением KDD является Знание знаний в базах данных.

В описывают процесс KDD

1. **Очистка данных**удалить шум и противоречивые данные.
2. **Интеграция данных**где несколько источников данных могут быть объединены.
3. **Выбор данных**где данные, относящиеся к задаче анализа, извлекаются из базы данных.
4. **Преобразование данных**где данные преобразуются и консолидируются в формы, подходящие для майнинга, путем выполнения операций сводки или агрегирования.
5. **Сбор данных**, который является важным процессом, где интеллектуальные методы применяются для извлечения шаблонов данных.
6. **Оценка картины**выявить действительно интересные модели, представляющие знания на основе интересных мер.
7. **Презентация знаний**где методы визуализации и представления знаний используются для представления добытых знаний пользователям.

В этой книге авторы отмечают, что интеллектуальный анализ данных чаще всего относится ко всему процессу обнаружения знаний из данных, возможно, потому, что это более короткий термин.

В этом разделе мы рассмотрим процесс обнаружения знаний в базах данных (KDD) в авторитетных статьях в этой области. Это обе статьи в виде презентабельных технических статей, а не рецензируемые статьи в журналах. Тем не менее, менее формальный тон позволяет провести полезное обсуждение этой темы высокого уровня.

От интеллектуального анализа данных до обнаружения знаний в базах данных

Эта статья была опубликована в 1996 году в журнале AI Magazine Усамой Файядом, Григорием Пятецким-Шапиро и Падрейк Смит.

Они определяют KDD как обнаружение знаний в базах данных, и это определение, с которым я более знаком:

«... Область KDD связана с разработкой методов и техник для понимания данных. ... В основе процесса лежит применение специальных методов интеллектуального анализа данных для обнаружения и извлечения шаблонов ».

а также

«... KDD относится к общему процессу получения полезных знаний из данных, а интеллектуальный анализ данных относится к определенному этапу этого процесса. Интеллектуальный анализ данных - это применение специальных алгоритмов для извлечения шаблонов из данных. »

Авторы предоставляют полезную сводку KDD на рисунке с сущностями в блоках и процессами, которые соединяют блоки как преобразования в сущностях. Это описание приводится ниже. Я стараюсь воспроизвести изображение, извините, официальные публикации могут быть трудными в этом отношении.

- **Шаг 1**: Выбор (данные в целевые данные)
- **Шаг 2**: Предварительная обработка (целевые данные в обработанные данные)
- **Шаг 3**: Преобразование (обработанные данные в преобразованные данные)
- **Шаг 4**: Интеллектуальный анализ данных (преобразование данных в шаблоны)
- **Шаг 5**: Интерпретация и / или оценка моделей в знания)

Этот процесс прост, и это модель, которую мне нравится использовать при работе над проблемой.

Процесс KDD для извлечения полезных знаний из объемов данных

Это была статья в Сообщениях ACM в 1996 году Усамы Файяда, Грегори Пиатецки-Шапиро и Падрейк Смит.

В этой статье авторы дают более подробное описание процесса KDD. Эта более подробная версия была в статье «Из Data Mining...» выше, но я чувствовал, что она была менее четко представлена. Это более подробное описание процесса KDD перефразировано ниже.

1. Понять предметную область и цель процесса
2. Создать целевой набор данных как подмножество всех доступных данных
3. Очистка и предварительная обработка данных для устранения шума, обработки недостающих данных и выбросов

4. Сокращение и прогнозирование данных, чтобы сосредоточиться на функциях, которые имеют отношение к проблеме
5. Сопоставьте цели процесса с методом интеллектуального анализа данных. Определите цель модели, такую как суммирование или классификация.
6. Выберите алгоритмы интеллектуального анализа данных, соответствующие цели модели (из шага 5)
7. Интеллектуальный анализ данных, то есть запуск алгоритмов на данных.
8. Интерпретация разработанных шаблонов, чтобы сделать их понятными для пользователя, такие как суммирование и визуализация.
9. Действовать на основании обнаруженных знаний, таких как отчетность или принятие решений.

Мне нравится деталь в этом процессе. Это действительно объясняет необходимость понимания целей процесса и выдерживания выбранного алгоритма в соответствии с этими целями.

Основы парадигмы MapReduce

Первое, что следует держать в голове, — в больших данных почти любая БД хранит данные на нескольких серверах.

Представим, что у нас есть 3 сервера и таблица клиентов. Вся таблица равномерна распределена на 3 части, каждый сервер хранит 1/3 данных. Чтобы вернуть результат запроса, нужно прочитать каждую часть с каждого сервера и собрать всё на одном сервере, где мы просматриваем результат.

Простейший SQL-запрос:

```
SELECT * FROM CLIENTS
```

Даже такой простой запрос разбивается на три обязательные части MapReduce:

1. Map
2. Shuffle
3. Reduce

Каждая из этих операций по-разному распределяется и параллелится, поэтому важно понимать, что они собой представляют.

1. Стадия Map зачастую представляет собой обычное чтение с жёсткого диска. Кроме чтения здесь могут применяться однострочные трансформации и фильтры, т.е. операции без join, group by, order by, distinct и без агрегирующих функций. Операции на этой стадии всегда хорошо параллелятся и не создают нагрузку на БД, т.к. каждый сервер читает только ту часть данных, которая имеется у него на жёстком диске. Данные зачастую сохранены равномерно на каждом сервере. Все серверы участвуют в этой стадии и делят нагрузку равномерно.
2. На стадии Shuffle никаких вычислений не происходит, зато все данные перемещаются между серверами таким образом, чтобы из них можно было получить финальный результат на стадии Reduce. Этот шаг станет понятен только после погружения в стадию Reduce, поэтому перейдём к ней.
3. Стадия Reduce самая коварная, т.к. она может провоцировать большие проблемы с производительностью БД. Здесь происходят все группирующие операции, а также операции, которые записывают результат. Некоторые операции не могут выполняться одновременно на нескольких серверах, поэтому для их выполнения требуется собрать весь объем данных на одном сервере. Если данные не помещаются на один сервер, запрос всегда будет выдавать ошибку.

Подробнее разберём на примерах ниже.

Как писать эффективные SQL-запросы

Вернемся к запросу:

```
SELECT * FROM CLIENTS
```

Здесь мы получим равномерное чтение таблицы на трёх серверах. Но что делать с результатом? Если мы захотим вывести его на экран, результат должен быть собран на один сервер, с которого мы выполняем запрос. Получается, что наш последний шаг вывода на экран сводит распределённые вычисления на нет — вся финальная нагрузка придётся на наш сервер, где мы получаем результат.

Стадия Map будет распределённой. Далее последует стадия Shuffle, которая перекинет все данные на один финальный сервер, который должен будет вместить весь результат и вывести его на экран. Если данные настолько большие, что не помещаются на один сервер, даже такой простейший запрос никогда не выполнится. Результирующий сервер всегда будет возвращать ошибку Out of memory.

Схема модели MapReduce

Этот запрос можно достаточно легко изменить:

```
INSERT INTO CLIENTS_NEW SELECT * FROM CLIENTS
```

Теперь вместо вывода результата на экран результат будет записан в другую таблицу. Поскольку другая таблица также хранится распределённо, запись могут производить одновременно 3 сервера.

Таким образом, не потребуется собирать все данные в одном месте, все вычисления будут хорошо распределяться. Стадия Map будет хорошо параллелиться, однако теперь стадия Reduce (запись результата) будет выполняться распределённо на тех же серверах, где данные и были прочитаны. Значит, мы можем пропустить стадию Shuffle (не передавать данные между серверами перед записью результата), что тоже ускоряет вычисления.

Данные не собираются на одном сервере

Аналогичная логика применима к операциям с фильтрами, такими как `SELECT * FROM CLIENTS WHERE CLIENTS.GENDER = 1` и так далее. Такие фильтры также будут выполняться распределённо на стадии Map.

Операции с агрегациями

Рассмотрим теперь операции с агрегациями. Допустим, мы хотим посчитать количество клиентов по полу.

SQL-запрос такой:

```
SELECT COUNT(*) FROM CLIENTS GROUP BY CLIENTS.GENDER
```

Стадия Map без сюрпризов, снова параллельное чтение тремя серверами. А вот Reduce всё меняет.

Поскольку у нас есть группировка по полу, в ответе мы хотим увидеть два числа — количество мужчин и количество женщин. Значит, на стадии Reduce мы можем задействовать максимум два сервера. Один сервер должен считать всех мужчин, другой — всех женщин. Для этого на стадии Shuffle необходимо передать записи всех мужчин на один сервер, а записи женщин — на другой сервер. Тогда на этапе Reduce результирующим серверам останется только посчитать все записи, полученные на этапе Shuffle.

Мы видим, что Reduce распределяет вычисления в зависимости от группирующих функций. Такая логика применяется для всех агрегатных функций, distinct, join (где группировка идет в зависимости от условия join) и для сортировок.

На стадии Shuffle данные разделяются по указанному признаку

С сортировкой нужно быть особенно аккуратным. Чтобы отсортировать все записи без группировки по ключу, Reduce соберёт все записи на один сервер и будет производить сортировку нераспределённо. Поэтому нужно избегать операций, которые выполняются с неравномерной группировкой (когда группирующих ключей меньше, чем доступных серверов).

Теперь вы знаете, на что нужно обращать внимание при написании запросов к большим данным. Ключ к написанию эффективного запроса — наблюдение за потребляемыми ресурсами БД в зависимости от изменения вашего запроса. Меняйте порядок join, группировок, подзапросы и ищете наилучшее сочетание производительности

1. Концепция машинного обучение

Машинное обучение – это научное исследование алгоритмов и статистических моделей, которые компьютерные системы используют для эффективного выполнения конкретной задачи без использования явных инструкций, опираясь на шаблоны и выводы. В более простых словах можно сказать что машинное обучение – это наука о том, как заставить компьютера учиться на основе их опыта без фактического традиционного программирование, то есть без какого-либо вмешательства человеческой помощи. Данная наука стремится ответить на вопрос “как мы, то есть люди, можем построить компьютерные системы, которые автоматически улучшаются с опытом. И каковы фундаментальные законы, которые управляют всеми процессами обучения?”. Так же можно отметить что цель данной науки – стремится к тому, чтобы сделать машиной “умной”. Так как на Земле самые умные существа – это люди, то можно смело сказать, что на данном этапе цель науки машинного обучение — это сделать машину с таким интеллектом, который будет максимально приближен к человеческому.

Абстрактный процесс машинного обучение начинается с подачи входных данных для алгоритма. Эти данные могут быть разными, начиная с более простых, такие как числовые или текстовые данные, заканчивая более сложными, например, видео файл или изображение. Эти данные считываются электронно-вычислительной машиной(компьютером) и в последовательности обрабатываются различными математическими и статистическими алгоритмами. После обработки компьютер выводит эти же данные в виде информации, которые могут быть полезны для анализа, принятие решении, иллюстрации и т.д. Весь процесс машинного обучение выглядит очень просто для человека, но почему тогда его считают наукой или областью научных исследований? На самом деле первый и последний этап — это просто управляющие элементы машинного обучение, а второй этап, то есть обработка входных данных и есть само машинное обучение. Почему считывание и вывод данных являются управляющими? Так как в машинном обучении — это статистика и математические алгоритмы, то в зависимости от типа данных и в как эти данные конечном итоге должны выглядит в виде информации и выбираются наиболее подходящие методы и алгоритмы обработки данных. То есть в зависимости цели и задачи и выбирается какой алгоритм и будет использоваться для решения конкретной задачи и цели.

Этап обработки, во всех случаях, кроме самых тривиальных, понимание или знание, которые вы пытаетесь получить из необработанных или сырых данных. Например, если задача состоит в том, чтобы написать обзор или рецензию на книгу, то человек не может это сделать, прочитав в книге пару ключевых слов или пару страниц. Для этого мы должны полностью прочитать данную книгу и после того как мы поймем полное содержание книги, мы можем изложить всю суть информации в кратком обзоре или рецензии. То есть книгу из двухсот или более страниц можно сжать до пару страниц, где есть самые нужные и главные идеи данной книги, и может считаться полноценной информацией. И цель использование машинного обучение состоит в том, чтобы превращает данные в информацию.

Машинное обучение лежит на стыке информатики, инженерии и статистики и часто появляется в других дисциплинах. Его можно применить ко многим областям – от политики до наук о земле. Это инструмент, который может быть применен ко многим проблемам. Любая область, которая должна интерпретировать и действовать на основе данных, может извлечь выгоду из методов машинного обучения. Машинное обучение использует статистику. Алгоритмы машинного обучения строят математическую модель на основе выборочных данных, известных как" обучающие данные", чтобы делать прогнозы или решения без явного программирования для выполнения задачи. Алгоритмы машинного обучения используются в самых разных приложениях, таких как фильтрация электронной почты и компьютерное зрение, где невозможно разработать алгоритм конкретных инструкций для выполнения задачи.

Но что именно означает машинное обучение для вычислительной машины (компьютера)? Митчелл Том в своих работах дал более формальное определение алгоритмам, используемых в машинном обучении: “ Говорят, что компьютерная программа обучается на основе опыта E по отношению к некоторому классу задач *T* и меры качества *P*, если качество решения задач из *T*, измеренное на основе *P*, улучшается с приобретением опыта *E*”. Для примера можно привести шахматы, где *E* это количество сыгранных партий в шахматах, *T* это задача играть в шахматы и победить партию, *P* вероятность того, что компьютер выиграет следующую партию. Машинное обучение будет увеличивать количество сыгранных партий – *E*, для того чтобы вероятность победы была ближе к абсолютной, то есть близка к 100 процентов. То есть машинное обучение должно “обучаться” на основе собственного опыта для решения задачи и цели и для увеличения точности выходных данных.

Преимущество машинного обучение над человеческим интеллектом состоит в том, что машина может находить скрытые связи, которые критично могут повлиять на результат выходных данных. То есть информация машинного обучение может отличаться от той информации который мы предполагаем получит. Данное преимущество очень полезно при обработке больших данных, где человеку очень трудно отыскать простые связи, не говоря о том, что есть и скрытые.

В большинстве случаев машинное обучение как раз и применяется для больших данных, для сокращения трудовых и материальных ресурсов.

2. Определение больших данных в машинном обучении

Большие данные – это любой источник данных, который имеет хотя бы одну из четырех общих характеристик:

1. Объем – количество генерируемых и хранимых данных. Размер данных определяет ценность и потенциальное знание, а также может ли он считаться большими данными или нет.

2. Многообразие – тип и характер данных. Большие данные могут быть из текста, изображения, аудио, видео.
3. Скорость – в этом контексте, скорость, с которой данные генерируются и обрабатываются для удовлетворения требовании и задач. Большие данные часто доступны в режиме реального времени. По сравнению с малыми данными, большие данные создаются непрерывно. Есть два вида скорости, связанные с большими данными – это частота генерации и частота обработки, записи и публикации.
4. Достоверность – это расширенное определение для больших данных, которое относится к качеству и значению данных. Качество данных может сильно варьироваться, влияя на конечный результат анализа или обработки.

В данное время точное определение больших данных нет. Так как большие данные эволюционировали так быстро и беспорядочно, что такого общепринятого формального утверждения, обозначающего их значение, не существует. Было много попыток определения больших данных, более или менее интересных с точки зрения использования и цитирования. Однако ни одно из этих предложений не помешало авторам работ, связанных с большими данными, расширить, обновить или даже игнорировать предыдущие определения и предложить новые. Хотя большие данные – это еще относительно молодая концепция, она, безусловно, заслуживает общепринятого словаря ссылок, который позволяет должным образом развивать дисциплину среди научного круга.

Но в машинном обучении большие данные обычно связывают с источниками данных, которые и используются для обучения компьютера, аналитики, обработки, классификации, кластеризации и других операций.

Традиционные данные используют централизованную архитектуру баз данных, в которой большие и сложные задачи решаются одной компьютерной системой. Централизованная архитектура является дорогостоящей и неэффективной для обработки большого объема данных. Большие данные основаны на распределенной архитектуре базы данных, где большой блок данных решается путем его разделения на несколько меньших размеров. Тогда решение проблемы определяется несколькими различными компьютерами в компьютерной сети. Компьютеры общаются друг с другом, чтобы найти решение проблемы. Распределенная база данных обеспечивает лучшие вычисления, более низкую цену и также улучшает представление по сравнению с централизованной системой базы данных. Это связано с тем, что централизованная архитектура основана на мэйнфреймах, которые не так экономичны, как микропроцессоры в распределенной системе баз данных. Также распределенная база данных обладает большей вычислительной мощностью по сравнению с централизованной системой баз данных, которая используется для управления традиционными.

Большие данные представляют собой общую область проблем и методов, используемых для доменов приложений, которые собирают и поддерживают огромные объемы необработанных данных для анализа данных по конкретным доменам. Развитию науки о больших данных в значительной степени способствовали

современные информационные технологии, а также увеличение вычислительных ресурсов и ресурсов хранения данных. Технологических компаний, таких как Google, Yahoo, и Microsoft, и Amazon хранят данные, которые измеряются в пропорции экзабайт или больше. Более того, в социальных сетях, таких как Facebook, YouTube и Twitter, миллиарды пользователей постоянно генерируют очень большое количество данных. Различные организации инвестировали в разработку продуктов с использованием аналитики Больших Данных для решения их мониторинга, экспериментов, анализа данных, моделирования и других потребностей в знаниях и бизнесе, что делает его центральной темой в исследованиях науки о данных.

Точность модели машинного обучения может значительно возрасти, если она обучается на больших данных. Без достаточного количества данных алгоритм машинного обучение будет пытаться принимать решения по небольшим подмножествам данных, которые могут привести к неправильной интерпретации тренда или отсутствию шаблона. Данные должны быть проверены на основе достоверности и контекста. Необходимо определить правильный объем и типы данных, которые могут быть проанализированы для влияния на результаты. Большие данные включает в себя все данные, включая структурированные, неструктурированные и полуструктурированные данные из электронной почты, социальных сетей, текстовых потоков, изображений и датчиков машины.

Машинное обучение требует правильного набора данных, которые могут быть применены в процессе обучения. Любой институт не должен хранить большие данные для использования методов машинного обучения, однако, большие данные могут помочь улучшить точность моделей машинного обучения. С большими данными теперь можно виртуализировать данные, чтобы их можно было хранить наиболее эффективным и экономичным образом, будь то хранилище данных или в облаке.

Извлечение значимых паттернов из массивных входных данных для принятия решений, прогнозирования и других выводов лежит в основе анализа Больших Данных. Помимо анализа больших объемов данных, анализ больших данных создает другие уникальные проблемы для машинного обучения и анализа данных, включая изменение формата необработанных данных, быстрое перемещение потоковых данных, достоверность анализа данных, сильно распределенные источники ввода, шумные и некачественные данные, высокую размерность, масштабируемость алгоритмов, несбалансированные входные данные, неконтролируемые и неклассифицированные данные, ограниченные контролируемые/помеченные данные и т.д. Адекватное хранение данных, индексирование/маркировка данных и быстрый поиск информации являются другими ключевыми проблемами в анализе больших данных. Следовательно, при работе с большими данными необходимы инновационные решения для анализа и управления данными.

В заключении можно сказать что большие данные и машинное обучение тесно связаны друг с другом, так как большие данные бесполезны без его анализа и извлечение информации, а машинное обучение не смогла бы сосуществовать без больших данных, которые дают алгоритму опыт и обучение.

3. Виды машинного обучения

Машинное обучение, как науку, можно классифицировать на 3 основные категории в зависимости от характера обучения:

1. обучение с учителем;
2. обучение без учителя;
3. обучение с подкреплением.

В некоторых научных работах по характеру обучение делят на 4 категории, где включают частичное обучение, но это всего лишь симбиоз обучение с учителем и без учителя.

3.1. Обучение с учителем

Это когда алгоритм учится на примерах данных и связанных целевых ответах, которые могут состоять из числовых значений или строковых меток, таких как классы или теги, чтобы позже предсказать правильный ответ, когда он задан с новыми примерами. Алгоритм предназначен, для того чтобы найти закономерности в данных, которые могут быть применены в процессе анализа. Эти данные имеют помеченные объекты, которые определяют значение данных. Этот подход действительно похож на обучение человека под руководством учителя. Учитель дает ученику хорошие примеры для запоминания, и затем ученик извлекает общие правила из этих конкретных примеров. Например, могут быть миллионы изображений животных и включать объяснение того, что такое каждое животное, а затем можно создать приложение машинного обучения, который отличает одно животное от другого. Обозначая эти данные о типах животных, вы можете иметь сотни категорий различных видов.

Для того, чтобы решить проблему с целью применения обучения с учителем, необходимо выполнить следующие шаги:

1. Определите тип обучающих примеров. Прежде всего, нужно решить какие данные должны использоваться в качестве обучающего набора.
2. Сбор данных. Набор данных должен быть репрезентативным для реального использования функции. Таким образом, собирается набор входных объектов и соответствующие выходные данные.
3. Определение входного представление объекта изучаемой функции. Точность изучаемой функции сильно зависит от того, как представлен входной объект. Как правило, входной объект преобразуется в вектор объектов, который содержит ряд

объектов, описывающих объект. Количество функций не должно быть слишком большим, из-за «проклятия размерности», но должен содержать достаточно информации, чтобы точно предсказать результат.

4. Сформулировать структуру изучаемой функции и соответствующий алгоритм обучения.
5. Завершение проектирование и архитектуры. Применение алгоритма обучения на собранном обучающем наборе. Некоторые контролируемые алгоритмы обучения требуют от пользователя определения определенных параметров управления. Эти параметры могут быть скорректированы путем оптимизации производительности подмножества (называемого набором проверки) обучающего набора или путем перекрестной проверки.
6. Оценка точности изученной функции. После настройки параметров и обучения, производительность функции должна измеряться на тестовом наборе, отдельном от обучающего набора данных.

Алгоритмы обучаются с использованием предварительно обработанных примеров, и на этом этапе производительность алгоритмов оценивается с помощью тестовых данных. Иногда шаблоны, идентифицированные в подмножестве данных, не могут быть обнаружены в большей совокупности данных. Если модель подходит только для представления шаблонов, существующих в подмножестве обучения, создается проблема под названием “Переобучение”.

Переобучение означает, что модель точно настроена для обучающего набора данных, но не может быть применима для больших наборов неизвестных данных. Для защиты от чрезмерной подгонки тестирование должно проводиться против непредвиденных или неизвестных данных. Использование непредвиденных данных для набора тестов может помочь оценить точность модели при прогнозировании результатов. Модели контролируемого обучения имеют широкую применимость к различным бизнес-проблемам, включая обнаружение мошенничества, рекомендации, распознавание речи или анализ рисков.

Наиболее широко используемые и популярные алгоритмы обучения с учителем:

- метод опорных векторов;
- линейная регрессия;
- логистическая регрессия;
- наивный байесовский классификатор;
- обучение дерева решений;
- метод k-ближайших соседей;
- искусственная нейронная сеть;
- изучение сходства.

Каждые вышеупомянутые алгоритмы имеют различные подходы математических и статистических математических методов, и формул. Но можно подчеркнуть общий шаблон алгоритма, так как все эти алгоритмы относятся к обучению с учителем:

Обучающий набор данных состоит из n упорядоченных пар $(x_1,y_1),(x_2,y_2),...,(x_n,y_n)$, где каждый x_i – измерение или набор измерений одной точки данных, а y_i – ответ для этой точки данных. Например, x_i может быть группой из пяти измерений для пациента в больнице, включая рост, вес, температуру, уровень сахара в крови и кровяное давление. Соответствующий y_i может быть

классификацией пациента как «здоровый” или „больной“.

Тестовые данные в контролируемом обучении – это еще один набор измерений m без ответа: *(xn+1, xn+2,..., xn+m) * (x_(n+1),x_(n+2),...,x_(n+m))*. Как описано выше, цель состоит в том, чтобы сделать обоснованные предположения ответа для тестовых данных(например, “здоровый” или “ больной ”), делая выводы из построенной шаблонной модели, который был создан при обработке обучаемых данных.

3.3. Обучение с подкреплением

Обучение с подкреплением – это поведенческая модель обучения. Алгоритм получает обратную связь от анализа данных и предсказание алгоритма, поэтому алгоритм ориентируется на оптимальный результат. Обучение с подкреплением отличается от других типов обучения с учителем, поскольку система не обучается с набором выборочных(обучающих) данных. Алгоритм учится методом проб и ошибок. Поэтому последовательность успешных решений приведет к тому, что процесс будет “подкреплен” и что он лучше всего будет решать проблему с каждым разом.

Алгоритм не знает какие действия предпринять, но вместо этого он должен выяснить, какие действия приносят наибольшую положительную оценку (вознаграждение), пробуя сопоставлять ситуации с действиями. В самых интересных и сложных случаях действия могут влиять не только на непосредственное оценку, но также и на следующую ситуацию и все последующие оценки, которые называются поздней оценки. Эти две характеристики – поиск методом проб и ошибок и поздняя оценка – являются двумя наиболее важными отличительными чертами обучения с подкреплением.

Одной из проблем, возникающих в процессе обучения, является компромисс между изучением и освоением. Чтобы получить хорошую оценку обратной связи, алгоритм обучающийся подкреплению, должен предпочесть те действия, которые он пробовал в прошлом и посчитал эффективными в получении хорошей обратной связи. Но, чтобы обнаружить такие действия, алгоритм должен попробовать новые действия, которые он не делал раньше. Алгоритм должен использовать то, что он уже испытал, чтобы получить оценку, но он также должен делать новые действия, чтобы сделать лучший выбор действий в будущем для новых данных.
Дилемма заключается в том, что ни изучение, ни освоение не могут осуществляться исключительно без сбоев в выполнении задачи. Алгоритм должен пробовать различные действия и постепенно отдавать предпочтение тем, которые кажутся лучшими. В стохастической задаче каждое действие должно быть многократно опробовано, чтобы получить надежную оценку. Дилемма изучение – освоение интенсивно изучалась математиками в течение многих десятилетий, но остается нерешенной.

Ошибки помогают учиться, потому что они добавляют меру взыскания (стоимость, потеря времени, сожаление, боль и т. д., уча вас, что определенный курс действий менее вероятен, чем другие. Интересный пример обучения подкреплению происходит, когда компьютеры учатся играть в видеоигры сами по себе без вмешательства людей.

Машинное обучение так же можно классифицировать на основе требуемых результатов:

1. классификация;
2. кластеризация;
3. регрессия.

Алгоритмы регрессии обычно используются для статистического анализа. Регрессия помогает анализировать модельные отношения между точками данных. Алгоритмы регрессии могут количественно определять силу корреляции между переменными в наборе данных. Кроме того, регрессионный анализ может быть полезен для прогнозирования будущих значений данных на основе исторических значений. Однако важно помнить, что регрессионный анализ предполагает, что корреляция связана с причинно-следственной связью. Без понимания контекста вокруг данных регрессионный анализ может привести к неточным прогнозам. Виды регрессии:

- линейная регрессия;
- нелинейная регрессия;
- векторная регрессия;
- логистическая регрессия.

Кластеризация – довольно простой метод для понимания. Объекты с аналогичными параметрами группируются вместе (в кластере). Все объекты в кластере более похожи друг на друга, чем на объекты других кластеров. Кластеризация – это тип обучение без учителя, поскольку алгоритм сам определяет общие характеристики элементов в данных. Алгоритм интерпретирует параметры, составляющие каждый элемент, а затем группирует их соответствующим образом.

Категории кластеризации:

- метод k-средних;
- основанная на плотности пространственная кластеризация для приложений с шумами – DBSCAN;
- алгоритм кластеризации OPTICS;
- метод главных компонент.

Но важно отметить, что при кластеризации, особенно в обучении без учителя, алгоритм ищет связи между входными данными. Прелесть машинного обучение – это поиск скрытых связей между данными, более известные как латентные связи. Для кластеризации при поиске латентных связей, используется модель скрытых переменных, который приминается для изучения взаимосвязей между значениями переменных. Модель скрытых переменных

включает в себя:

- EM-алгоритм;
- метод моментов;
- слепое разделение сигнала;
- метод главных компонент;
- анализ независимых компонент;
- неотрицательное матричное разложение;
- сингулярное разложение.

Классификация – это процесс прогнозирования класса заданных точек данных. Классы иногда называются метками или категориями. Классификационное прогнозирующее моделирование представляет собой задачу аппроксимации функции отображения *(f)* от входных переменных *(X)* к дискретным выходным переменным *(y)*. Классификация относится к категории обучение с учителем. Виды классификационных схем:

- тезаурус;
- таксономия;
- модель данных;
- транспортная сеть;
- онтология.

Но в машинном обучении виды классификации делаться по типам алгоритмов, которые так или иначе относится к схемам классификации. Наиболее широко используемые алгоритмы обучения:

- метод опорных векторов;
- логистическая регрессия;
- наивный байесовский классификатор;
- метод k-ближайших соседей;
- искусственная нейронная сеть;
- дерево принятия решений.

4. Машинное обучение для анализа текстовых данных

Анализе текстовых данных в машинном обучении используется методы регрессии, классификации и кластеризации. Данные методы были описаны в этой работе ранее. Но стоит отметить что есть главная отличие в анализе текстовых данных, так как сама обработка текста является очень сложной задачей в машинном обучении. Главная отличие – это интеллектуальный анализ текстовых данных. Так как текстовый документ

для человека – это набор слов, который несет смысл, для машины – это просто битовые данные. И задача интеллектуального анализа текстовых данных состоит в том, чтобы машина смогла понимать смысл текстового документа. Перед тем как использовать алгоритмы машинного обучение, нужно также применить методы обработки текстовых данных.

Обработка естественного языка (NLP) – интеллектуальный анализ данных и методы машинного обучения используются вместе для автоматической классификации и обнаружения шаблонов из электронных документов. Основная цель анализа текста – дать пользователям возможность извлекать информацию из текстовых ресурсов и заниматься такими операциями, как извлечение, классификация (машинное обучение) и суммирование. Анализ текста состоит из нескольких задач, таких как правильная аннотация к документам, соответствующее представление документа, уменьшение размерности для обработки алгоритмических вопросов и соответствующая функция классификатора для получения хорошего обобщения и избежание чрезмерной подгонки. Извлечение, интеграция и классификация электронных документов из различных источников и обнаружение знаний из этих документов имеют важное значение для исследовательских сообществ.

Процесс предварительной обработки состоит в том, чтобы очистить границу каждой языковой структуры и максимально устранить языковые факторы, зависящие от языка.

Представление документов является одним из методов предварительной обработки, который используется для уменьшения сложности документов и облегчения их обработки, документ должен быть преобразован из полнотекстовой версии в вектор документа. Текстовое представление является важным аспектом в классификации или кластеризации документов, обозначает отображение документа в компактную форму его содержания. Текстовый документ обычно представляется в виде вектора весов терминов (словарных признаков) из набора терминов (словаря), где каждый термин встречается хотя бы один раз в определенном минимальном количестве документов. Основной характеристикой проблемы классификации/кластеризации текста является чрезвычайно высокая размерность текстовых данных. Количество потенциальных возможностей часто превышает количество документов. Определение документа состоит в том, что он состоит из терминов, которые имеют различные шаблоны возникновения. Предварительная обработка включают в себя такие этапы как: извлечение признаков (характеристик) и выбор признаков.

Извлечение признаков является первым этапом предварительной обработки, который используется для представления текстовых документов в формате слов. Таким образом, алгоритмы удаление стоп-слов, стеммитизация, токенизация и другие – это задачи предварительной обработки. Данные алгоритмы относится к методу обработки естественного языка (NLP). NLP – может быть определена как автоматическая (орсеми-автоматическая) обработка человеческого языка. Термин NLP используется гораздо более узко, часто исключая поиск информации, а иногда даже исключая машинный перевод. В настоящее время NLP, по сути, является междисциплинарным: оно тесно связано с лингвистикой и машинным обучением. Он также имеет связи с

исследованиями в когнитивной науке, психологии, философии и математике (особенно логике). В машинном обучении NLP используется как раз для того, чтобы машина могла понимать тексты написанное на естественном языке. Есть несколько алгоритмов NLP, который может использоваться вместе, по отдельности или последовательно. В этой работе будет рассматриваться основные алгоритмы, которые используется именно в машинном обучении:

- **Токенизация** – лексический (семантический) анализ текста, который находит минимальную единицу в тексте. Минимальная единица – это токен. В лексическом анализе токеном может быть одно слово, предложение или абзац. В этой работе мы будем использовать токен как одно слово .
- **Частота термина в документе** – в семантическом анализе часто используют алгоритм счетчика слов, чтобы найти частоту каждого слова в данном тексте. Зачем эта техника нужна? Для того чтобы найти ключевые слова или уникальные термины, классифицировать или кластеризировать большой массив (корпус) документов, найти сходство корпуса и т. д. Распределение частот слов является фундаментальным фенотипом языка. Статистики и лингвисты изучали распределение частот слов, поскольку статистика использования слов дает ценное представление о языке, его конструкции и эволюции. Эти распределения давно изучаются за пределами статистики и лингвистики.
- **Стеммитизация** – это преобразование морфологических форм слов в его корень, при помощи удаление окончание морфологических преобразований. Корень не обязательно должен быть существующим словом в словаре, но все его варианты должны соответствовать этой форме после завершения алгоритма. Есть два момента, которые следует учитывать при использовании стеммера:
    - *Морфологические формы слова, как предполагается, имеют одно и то же базовое значение и, следовательно, должны быть сопоставлены с одним и тем же корнем.*
    - *Слова, которые не имеют одного и того же значения должны храниться отдельно*

Эти два правила достаточно хороши, пока найденные корни полезны для семантического анализа текста или обработки языка. Для языков с относительно простой морфологией влияние морфологического преобразование меньше, чем для языков с более сложной морфологией.

- **Стоп-листинг** – применяется для удаления стоп-слов и отдельных символов, которые появляются в текстовом документе, и, вероятно, частота стоп-слов может быть больше, чем другие ключевые слова в этом тексте. Вот почему этот шаг необходимо использовать для получения более четкого результата в машинном обучении.
- **Лемматизация** – алгоритм аналогичный к стеммизации. Различие в том, что лемматизация принимает во внимание морфологический анализ слов. Для этого необходимо иметь подробные словари, которые алгоритм может просмотреть, чтобы связать форму с ее леммой (инфинитив слово).

После извлечения признаков важным шагом предварительной обработки текста является выбор признаков для построения векторного пространства, что повышает масштабируемость, эффективность и точность текстового классификации или кластеризации. В целом, хороший метод выбора признаков должен учитывать характеристики входных данных и алгоритма. Основная идея выбора признака заключается в выборе подмножества объектов из исходных документов. Выбор признаков выполняется путем сохранения слов с наивысшим весом в соответствии с заданной мерой важности слова. Выбранные функции сохраняют первоначальный смысл и обеспечивают лучшее понимание данных и процесса анализа. Для классификации или кластеризации текста основной проблемой является высокая размерность пространства объектов. Почти каждый текстовый документ имеет большое количество признаков, большинство из которых не актуальны и полезны для задачи машинного обучение, и даже некоторые шумовые признаки могут резко снизить точность алгоритма. Поэтому выбор признака обычно используется для уменьшения размерности пространства признаков и повышения эффективности и точности машинного обучение.

В машинном обучении текстовый документ может частично совпадать со многими категориями в классификации или со многими кластерами в кластеризации. Наиболее часто используемы алгоритмы выборки признаков:

- **Частота термина–обратная частота документа (TF-IDF)** обычно используется для взвешивания каждого слова в текстовом документе в соответствии с его уникальностью. Вес слова (токена) часто используется для поиска информации и семантического анализа текста. Этот вес является статистической мерой, используемой для оценки того, насколько важно слово для документа в коллекции или корпусе Другими словами, подход TF-IDF отражает релевантность слов, текстовых документов и конкретных категорий.
- **Word2Vec** – это инструмент (набор алгоритмов) для вычисления векторных представлений слов, реализующий две основные архитектуры — непрерывный пакет слов (CBOW) и Скип-грамм. В качестве входных данных передается текстовый документ или слово, а выходные данные будут представлены как векторные переменные (координаторы в векторном пространстве).

2. Лабораторная часть

Министерство образования республики Беларусь

Учреждение образования
«Гомельский государственный университет
имени Франциска Скорины»

В.Н. ЛЕВАНЦОВ

МЕТОДЫ ОБРАБОТКИ БОЛЬШИХ МАССИВОВ ДАННЫХ **x**

Практическое пособие
по выполнению лабораторных работ

Гомель 2023

Введение

В настоящее время информационные системы, применяющие базы данных, представляют собой одну из важнейших областей современных компьютерных технологий. При построении корпоративных систем обработки данных формируется единое информационное пространство, работа в котором носит распределенный характер. Распределенная обработка данных реализуется в компьютерных сетях и требует определенной дисциплины взаимодействия. Общепринятым стандартом такого взаимодействия стала технология клиент-сервер, когда часть функций прикладной программы реализована на программе-клиенте, другая – на программе-сервере.

Серверы баз данных являются наиболее эффективным инструментом для создания приложений, оперирующими большими объемами информации и являются важнейшим звеном в их построении по схеме клиент-сервер. При этом эффективно реализуется интегрированность базы данных, то есть возможность одновременного доступа к данным из нескольких приложений. Создание клиент-серверного приложения, работающего с базой данных, требует прохождения следующих этапов:

1 – разработка структуры реляционной базы данных;

2 – администрирование базы данных на стороне сервера;

3 – программирование на стороне сервера;

4 – программирование на стороне клиента.

В данном методическом руководстве рассматриваются все этапы построения клиент-серверного приложения на примере создания системы делопроизводства деканата вуза. В настоящий момент, одной из самых популярных серверных платформ, является Microsoft SQL Server, на котором и построена рассматриваемая база данных. Клиентское приложение строится на основе СУБД Microsoft Visual Foxpro. Foxpro является специализированным языком программирования, который ориентирован на работу с базами данных. Эта программная среда позволяет создавать эффективные приложения, работающие с локальными, сетевыми и удаленными базами данных на разных серверных платформах, в том числе и на Microsoft SQL Server.

В предлагаемой работе ввод данных и их программирование осуществляется и на стороне клиента, и на стороне сервера. Используются все средства, входящие в состав Visual Foxpro для формирования клиент-серверного приложения и взаимодействия с сервером баз данных на основе Microsoft SQL Server. Взаимодействие клиентской и серверной части разрабатываемой базы данных построено таким образом, чтобы продемонстрировать основные возможности Microsoft SQL Server по управлению входящими потоками информации:

- с помощью триггеров, программируемых на стороне сервера, повышается скорость обработки входящих данных и снижается трафик в сети организации;

- разработка транзакций на сервере позволяет производить откат некорректно выполненных операций;

- с помощью средств мониторинга и сопровождения SQL Server решается ряд задач по администрированию сервера.

Таким образом целью данной работы является рассмотрение основных возможностей Microsoft SQL Server по управлению базой данных и решение основных задач по администрированию сервера, построение клиентских приложений для серверных платформ и организация взаимодействия сервера с клиентским программным обеспечением.

Данное методическое руководство разработано для «специальности» «АСОИ» и предназначено для изучения курса «Б и БД».

Лабораторная работа №1 Разработка базы данных на Microsoft SQL Server

Разработка клиент-серверной информационной системы начинается с разработки базы данных на стороне сервера и настройки серверной платформы. Здесь можно выделить следующие задачи:

1 – создание базы данных и установка ее свойств;

2 - разработка таблиц;

3 – установление отношений между таблицами и обеспечение целостности данных;

4 – программирование на стороне сервера, написание триггеров и транзакций;

5 - ввод первоначальных данных.

1.1. Создание базы данных и установка ее свойств.

Создайте новую базу данных.

Откройте Enterprise Manager, найдите в консоли папку Databases на сервере, который вы используете. SQL Server отобразит список баз данных (рис.1.1).

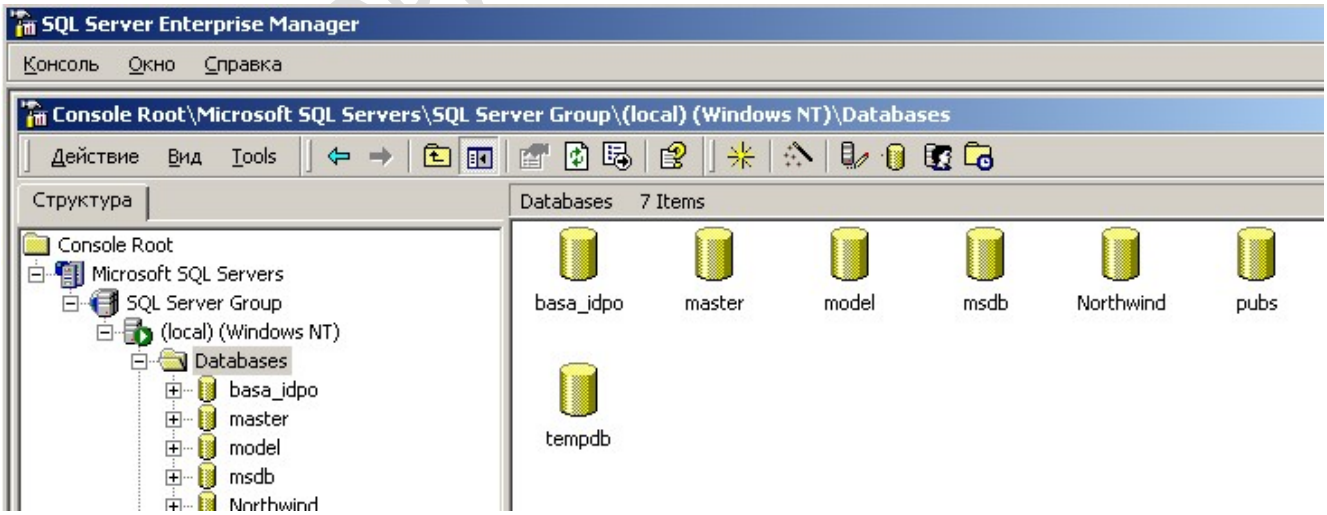


Рис.1.1 Создание базы данных

Выберите инструмент «New Database» и введите имя создаваемой базы данных.

Размещение файлов базы данных.

В процессе эксплуатации базы данных неоднократно возникают задачи переноса базы данных с одного носителя на другой и управления производительностью работы сервера. При создании новой базы данных формируются два файла ее сопровождения:

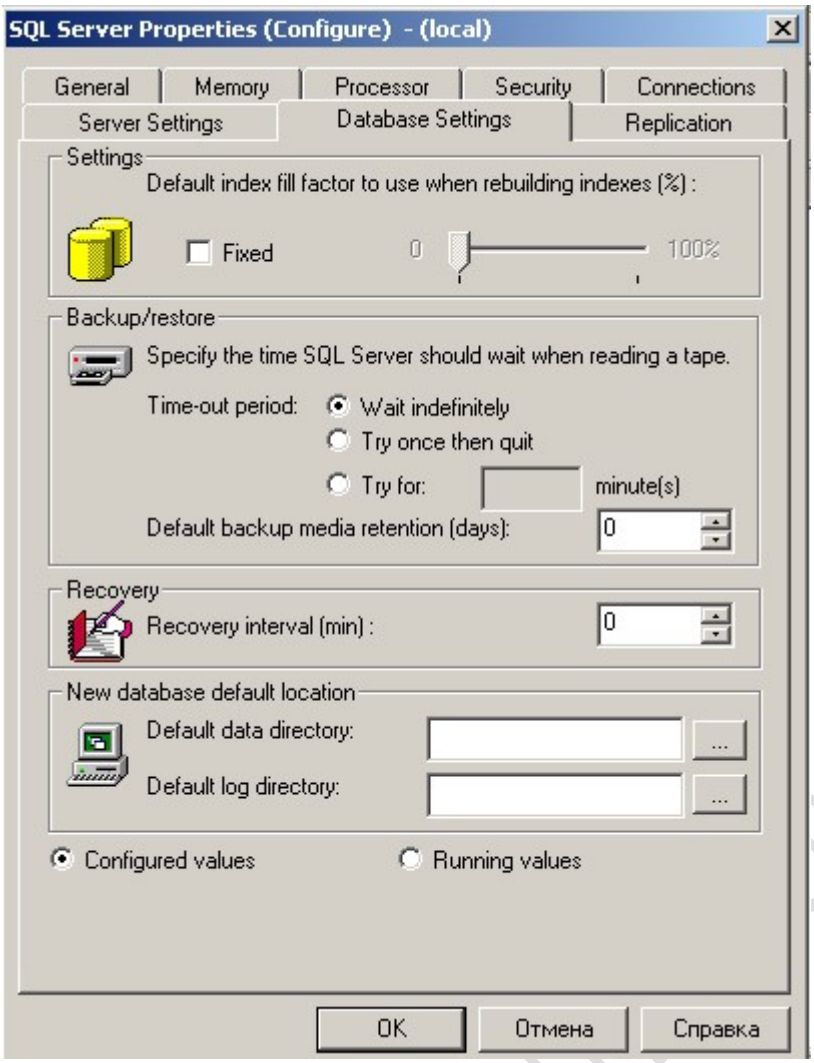
*.MDF (Master Data File)– файл данных, предназначен для хранения информации, находящейся в таблицах базы данных. Кроме того, в этом файле так же размещены процедуры, ограничения, триггеры, индексы и другая информация;

*.LDF – файл журнала транзакций, хранит информацию о ходе выполнения транзакций. В нем размещается информация о состоянии данных перед началом транзакции, о выполняемых изменениях, блокированных ресурсах и другая сопутствующая информация.

Любая база данных должна содержать как минимум один файл данных и один файл журнала транзакций. При необходимости администратор может добавлять в БД новые файлы данных или файлы журнала транзакций. Если компьютер, на котором установлен SQL Server имеет несколько физических дисков, то для повышения производительности Microsoft настоятельно рекомендует для каждой базы данных создать как минимум один файл на каждом физическом диске. Кроме того, по возможности следует располагать файлы данных и журнала транзакций на отдельных физических дисках. Это повышает производительность работы всего сервера баз данных.

При первой установке SQL Server для новых баз данных по умолчанию принимается место размещения: \диск установки SQL Server\Program Files\Microsoft SQL Server*MSSQL\data*. Эта настройка по умолчанию предоставляется мастеру создания базы данных Create Database Wizard. Чтобы изменить эту используемую по умолчанию установку, вы можете задать новое место размещения на вкладке Database Settings (Параметры базы данных) в диалоговом окне SQL Server Properties (Свойства SQL Server). Для этого:

1. Щелкните правой кнопкой мыши на сервере в дереве консоли Console Tree, выберите Properties (Свойства), а за тем откройте вкладку Database Settings (Параметры базы данных).



2. Перейдите к разделу Default data directoty и щелкните на кнопке Browse (Обзор), чтобы изменить местоположение файла базы данных. Мастер отобразит диалоговое окно, запрашивающее новое место размещения. Укажите нужную вам папку для размещения файла базы данных.

3. Перейдите к разделу Default log directoty и щелкните на кнопке Browse (Обзор), чтобы изменить местоположение файла журнала транзакций. Мастер отобразит диалоговое окно, запрашивающее новое место размещения. Укажите нужную вам папку для размещения файла базы данных.

Установка свойств базы данных.

Когда вы создаете базу данных с помощью мастера Create Database Wizard, вы указываете определенные характеристики, или свойства, базы данных, такие как имя базы данных и место

размещения. После создания базы данных вы можете поменять эти свойства, изменив соответствующие параметры в диалоговом окне Properties (Свойства).

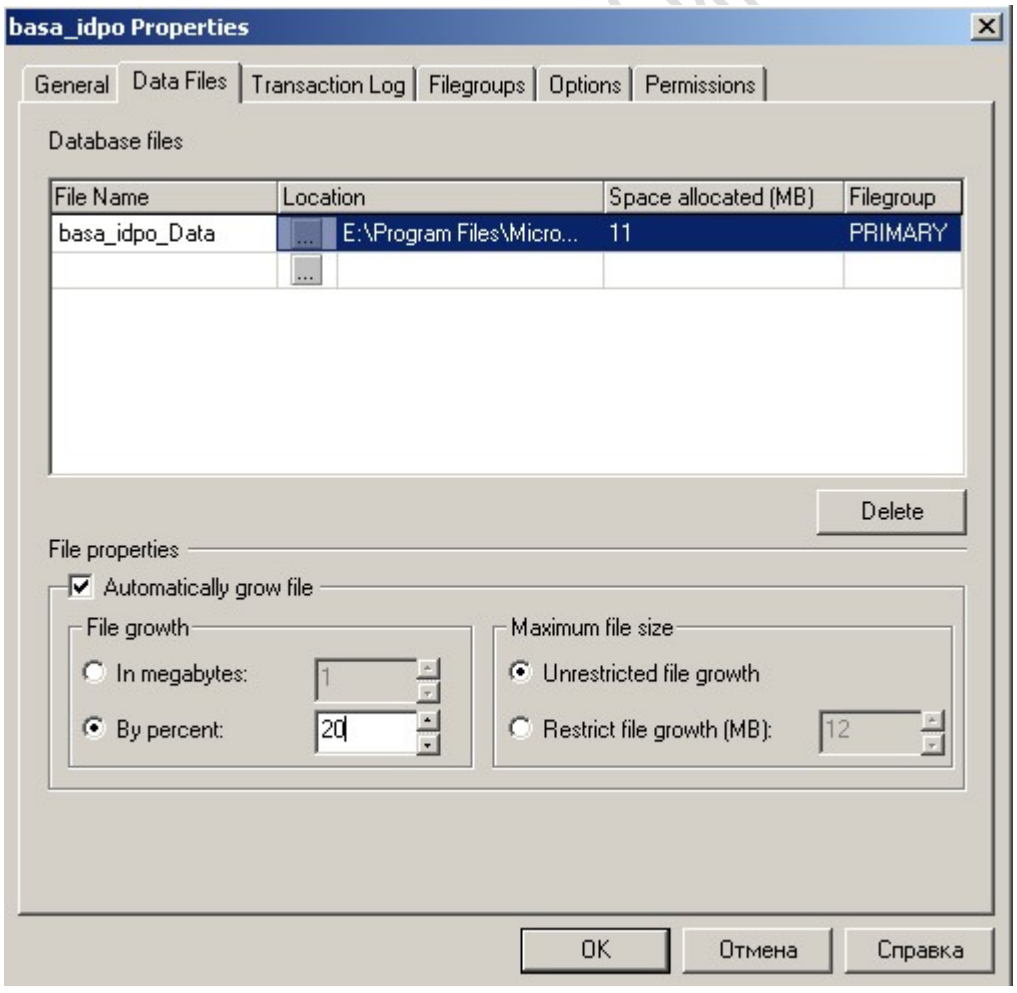
Например, увеличение размера физического файла — это довольно серьезная операция, выполнение которой может привести к увеличению времени отклика сервера. Если вы обнаружите, что SQL Server приходится слишком часто увеличивать размер файла. Вам следует учитывать возможность изменения процента увеличения размера файла в диалоговом окне Properties (Свойства), чтобы сервер смог увеличивать размер файла более чем на 10% (установка по умолчанию).

Изменение процента увеличения размера файла.

1. Выберите свою базу данных в дереве консоли Console Tree.

2. Нажмите кнопку Properties (Свойства) в панели инструментов. SQL Server отобразит диалоговое окно Properties (Свойства) для базы данных.

3. Откройте вкладку Data Files (Файлы данных). SQL Server отобразит свойства файлов данных базы данных.
4. Установите процент увеличения 20%.



5. Нажмите OK. SQL Server установит новое свойство и закроет диалоговое окно Properties (Свойства).

1.2. Создание таблиц в SQL Server.

Разработайте следующие таблицы, используя Enterprise Manager (рис.1):

студенты

	Column Name	Data Type	Length	Allow Nulls
🔑	id_студент	int	4	
	номер_дела	varchar	20	✓
	фамилия	varchar	20	✓
	имя	varchar	20	✓
	отчество	varchar	20	✓
	FK_специальность	int	4	✓
	FK_курс	int	4	✓
	FK_поток	int	4	✓
	место_работы	varchar	100	✓
	год_рождения	smalldatetime	4	✓
	соц_положение	varchar	15	✓
	адрес	varchar	80	✓
	образование	varchar	100	✓
	приказ_зачисления	varchar	30	✓
	дата_значения	smalldatetime	4	✓
	основание_зачисления	varchar	60	✓
	приказ_отчисления	varchar	30	✓
	дата_отчисления	smalldatetime	4	✓
	причина_отчисления	varchar	60	✓
	приказ_восстановления	varchar	30	✓
	дата_восстановления	smalldatetime	4	✓
	приказ_академа	varchar	30	✓
	дата_академа	smalldatetime	4	✓
	причина_академа	varchar	60	✓
	приказ_доплома	varchar	30	✓
	дата_диплома	smalldatetime	4	✓
	тема_диплома	varchar	250	✓
	оценка_диплома	int	4	✓
	квалификация	varchar	40	✓
	приказ_2_курс	varchar	10	✓
	дата_2_курс	smalldatetime	4	✓
	приказ_3_курс	varchar	10	✓
	дата_3_курс	smalldatetime	4	✓

дисцип_препод

	Column Name	Data Type	Length	Allow Nulls
🔑	id_дисцип_препод	int	4	
	FK_преподаватель	int	4	✓
	FK_предмет	int	4	✓
	FK_специальность	int	4	✓

учебный_план

	Column Name	Data Type	Length	Allow Nulls
🔑	id_уч_план	int	4	
	шифр_спец	varchar	50	✓
	FK_специальность	int	4	✓
	FK_дисциплина	int	4	✓
	FK_курсовая	int	4	✓
	FK_контрольная	int	4	✓
	FK_форма_контр	int	4	✓
	шифр_предмет	varchar	20	✓
	итого	int	4	✓
	лекции	int	4	✓
	практика	int	4	✓
	сам_работа	int	4	✓
	семестр	varchar	10	✓
	номер	int	4	✓

преподаватели

	Column Name	Data Type	Length	Allow Nulls
🔑	id_преподаватель	int	4	
	фамилия	varchar	50	✓
	имя	varchar	50	✓
	отчество	varchar	50	✓
	e_mail	varchar	50	✓
	место_работы	varchar	50	✓
	уч_степень	varchar	50	✓
	звание	varchar	50	✓
	фото	varchar	50	✓

успеваемость_студ

	Column Name	Data Type	Length	Allow Nulls
🔑	id_усп	int	4	
	ведомость	varchar	50	✓
	FK_спец	int	4	✓
	FK_курс	int	4	✓
	FK_поток	int	4	✓
	FK_предмет	int	4	✓
	FK_препод	int	4	✓
	семестр	int	4	✓
	часы	int	4	✓
	дата	smalldatetime	4	✓

оценка_студ

	Column Name	Data Type	Length	Allow Nulls
🔑	id_оценка	int	4	
	FK_усп	int	4	✓
	FK_студ	int	4	✓
	FK_оценка	int	4	✓
	FK_контр	int	4	✓
	FK_курс	int	4	✓
	FK_форма_контр	int	4	✓

специальности

	Column Name	Data Type	Length	Allow Nulls
🔑	id_спец	int	4	
	специальность	varchar	50	✓

курсовая

	Column Name	Data Type	Length	Allow Nulls
🔑	id_курс	int	4	
	курсовая	varchar	15	✓

предметы

	Column Name	Data Type	Length	Allow Nulls
🔑	id_предмет	int	4	
	предмет	varchar	150	✓

курс

	Column Name	Data Type	Length	Allow Nulls
🔑	id_курс	int	4	
	курс	varchar	10	✓

оценка

	Column Name	Data Type	Length	Allow Nulls
🔑	id_оценка	int	4	
	оценка	varchar	20	✓

контрольная

	Column Name	Data Type	Length	Allow Nulls
🔑	id_кр	int	4	
	контрольная	varchar	15	✓

поток

	Column Name	Data Type	Length	Allow Nulls
🔑	id_поток	int	4	
	поток	varchar	10	✓

ф_контроля

	Column Name	Data Type	Length	Allow Nulls
🔑	id_ф_контроля	int	4	
	контроль	varchar	50	✓

Рис.1.1. Таблицы базы данных

1.3. Создание связей между таблицами (рис.1.2)

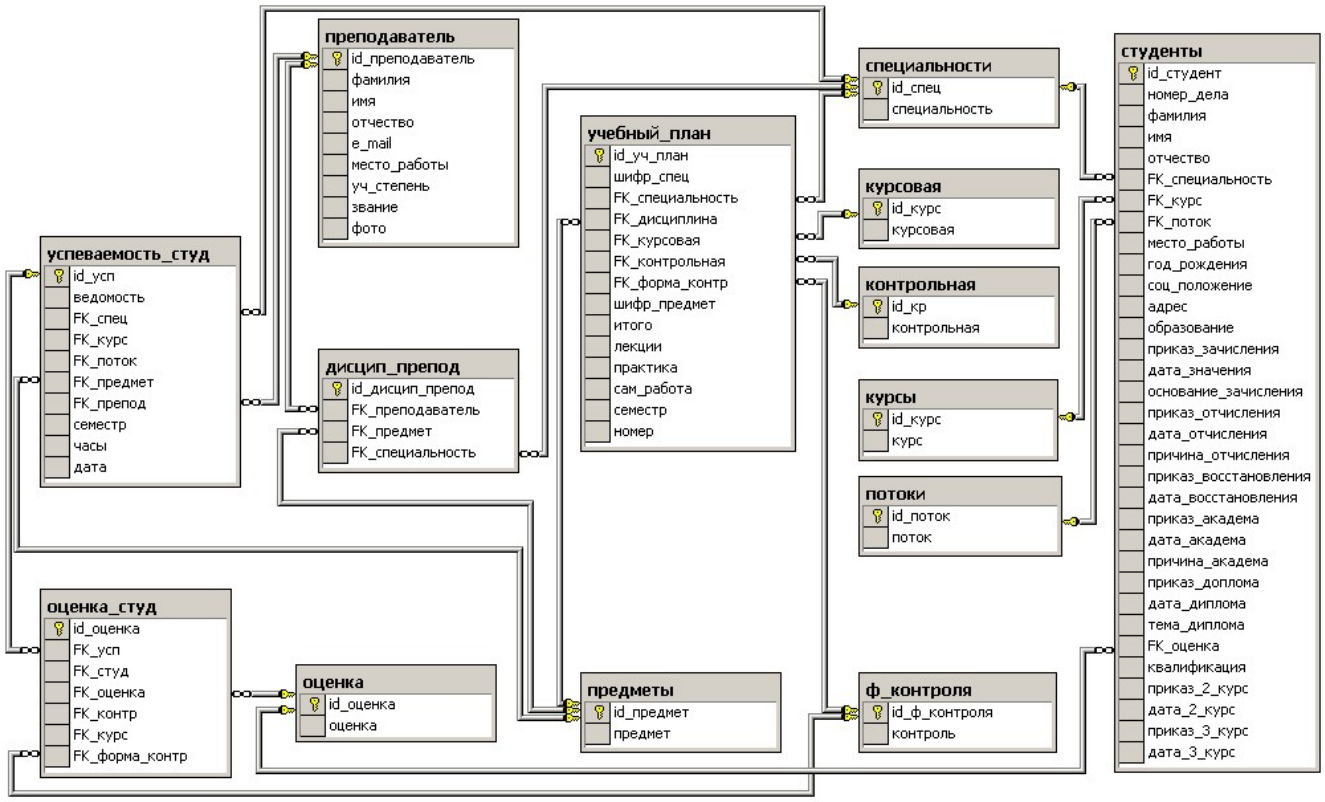


Рис.1.2. Схема базы данных

Таблицы связаны по следующим полям:

1- таблица «преподаватель», поле «id_преподаватель» – таблица «успеваемость_студ», поле «FK_препод»;

2- таблица «преподаватель», поле «id_преподаватель» – таблица «дисцип_препод», поле «FK_преподаватель»;

3- таблица «успеваемость_студ», поле «id_усп» – таблица «оценка_студ», поле «FK_усп»;

4- таблица «предметы», поле «id_ предмет» – таблица «учебный_план», поле «FK_дисциплина»;

5- таблица «предметы», поле «id_ предмет» – таблица «дисцип_препод», поле «FK_предмет»;

6- таблица «специальности», поле «id_спец» – таблица «учебный_план», поле «FK_специальность»;

7- таблица «специальности», поле «id_ спец» – таблица «дисцип_препод», поле «FK_специальность»;

8- таблица «специальности», поле «id_ спец» – таблица «студенты», поле «FK_специальность»;

9- таблица «потоки», поле «id_ поток» – таблица «студенты», поле «FK_поток»;

10- таблица «курсы», поле «id_ курс» – таблица «студенты», поле «FK_курс»;

11- таблица «оценка», поле «id_ оценка» – таблица «студенты», поле «FK_оценка»;

12 - таблица «предметы», поле «id_ предмет» – таблица «успеваемость_студ», поле «FK_предмет»;

13 - таблица «специальности», поле «id_ спец» – таблица «успеваемость_студ», поле «FK_спец»;

14- таблица «оценка», поле «id_ оценка» – таблица «оценка_студ», поле «FK_оценка».

Лабораторная работа №2 Создание запросов и модификация таблиц базы данных.

Цель работы: Используя данные базы данных, подготовленной в предыдущей лабораторной работе, подготовить и реализовать серию запросов, связанных с выборкой информации и модификацией данных таблиц.

Содержание работы и методические указания к ее выполнению

1. Изучить набор команд языка SQL, связанный с созданием запросов, добавлением, модификацией и удалением строк таблицы:

select - осуществление запроса по выборке информации из таблиц базы данных;

insert - добавление одной или нескольких строк в таблицу;

delete - удаление одной или нескольких строк из таблицы;

update - модификация одной или нескольких строк таблицы;

union - объединение запросов в один запрос.

2. Изучить состав, правила и порядок использования ключевых фраз оператора select:

select - описание состава данных, которые следует выбрать по запросу (обязательная фраза);

from - описание таблиц, из которых следует выбирать данные (обязательная фраза);

where - описание условий поиска и соединения данных при запросе;

group by - создание одной строки результата для каждой группы (группой называется множество строк, имеющих одинаковые значения в указанных столбцах);

having - наложение одного или более условий на группу;

order by - сортировка результата выполнения запроса по одному или нескольким столбцам;

into outfile - создание файла, в который будет осуществлен вывод результатов соответствующего запроса.

Порядок следования фраз в команде select должен соответствовать приведенной выше последовательности. Для лучшего понимания механизма функционирования выполните следующие упражнения:

I. Простые запросы на языке SQL

Запрос на языке SQL формируется с использованием оператора Select. Оператор Select используется

- для выборки данных из базы данных;
- для получения новых строк в составе оператора Insert;
- для обновления информации в составе оператора Update.

В общем случае оператор Select содержит следующие семь спецификаторов, расположенных в операторе в следующем порядке:

- спецификатор Select;
- спецификатор From;
- спецификатор Where;
- спецификатор Group by;
- спецификатор Having;
- спецификатор Order by;

Обязательными являются только спецификаторы Select и From. Эти два спецификатора составляют основу каждого запроса к базе данных, поскольку они определяют таблицы, из которых выбираются данные, и столбцы, которые требуется выбрать.

Спецификатор Where добавляется для выборки определенных строк или указания условия соединения. Спецификатор Order by добавляется для изменения порядка получаемых данных. Спецификатор Into temp добавляется для сохранения этих результатов в виде таблицы с целью выполнения последующих запросов. Два дополнительных спецификатора оператора Select - Group by (спецификатор группирования) и Having (спецификатор условия выборки группы) - позволяют выполнять более сложные выборки данных.

У п р а ж н е н и я

1. Выбор всех строк и столбцов таблицы.

Пример.

Выдать полную информацию о поставщиках.

Select * from S

Результат: таблица S в полном объеме.

Подготовьте запрос и проверьте полученный результат.

2. Измение порядка следования столбцов.

Пример.

Выдать таблицу S в следующем порядке: фамилия, город, рейтинг, номер_поставщика.

Select фамилия, город, рейтинг, номер_поставщика from S

Результат: таблица S в требуемом порядке.

Подготовьте запрос и проверьте полученный результат.

3. Выбор заданных столбцов.

Пример.

Выдать номера всех поставляемых деталей.

Select номер_детали from SPJ

Результат: столбец номер_детали таблицы SPJ

Подготовьте запрос и проверьте полученный результат.

4. Выбор без повторения.

Пример.

Выдать номера всех поставляемых деталей, исключая дублирование.

Select distinct номер_детали from SPJ

Результат:	**номер_детали**
	P1
	P2
	P3
	P4
	P5
	P6

Подготовьте запрос и проверьте полученный результат.

5. Использование в запросах констант и выражений.

Пример.

Select номер_детали, "вес в граммах", вес*454 from P

Результат:	P1 вес в граммах=5448

	P6 вес в граммах=8226

Подготовьте запрос и проверьте полученный результат.

6.Ограничение в выборке.

Пример.

Выдать номера всех поставщиков, находящихся в Париже с рейтингом > 20.

Select номер_поставщика from S where город="Париж" and рейтинг>20

Результат:	номер_поставщика
	S3

Подготовьте запрос и проверьте полученный результат.

7. Выборка с упорядочиванием.

Пример.

Выдать номера поставщиков, находящихся в Париже в порядке убывания рейтинга.

Select номер_поставщика, рейтинг from S where город="Париж" order by рейтинг desc

Результат:	номер_поставщика	рейтинг
	S3	30

	S2	10

Подготовьте запрос и проверьте полученный результат.

8. Упорядочивание по нескольким столбцам.

Пример.

Выдать список поставщиков, упорядоченных по городу, в пределах города - по рейтингу.

Select * from S order by 4, 3

Результат:	Номер_поставщика	Фамилия	Рейтинг	Город
	S5	Адамс	30	Атенс
	S1	Смит	20	Лондон
	S4	Кларк	20	Лондон
	S2	Джонс	10	Париж
	S3	Блейк	30	Париж

Подготовьте запрос и проверьте полученный результат.

9. Фраза in (not in).

Пример.

Выдать детали, вес которых равен 12, 16 или 17.

Select номер_детали, название, вес from P where вес in (12, 16, 17)

Результат:	номер_детали	Название	вес
	P1	Гайка	12
	P2	Болт	17
	P3	Винт	17
	P5	Кулачок	12

Подготовьте запрос и проверьте полученный результат.

12. Выбор по шаблону.

Для запросов с поиском по шаблону, основанных на поиске подстрок в полях типа CHARACTER, используются ключевое слова LIKE.

Включение в выражение ключевого слова NOT порождает условие с обратным смыслом. Ключевое слово LIKE соответствует стандарту ANSI.

СИМВОЛ	ЗНАЧЕНИЕ
LIKE	
%	Заменяет последовательность символов
-	Заменяет любой одиночный символ
\	Отменяет специальное назначение следующего за ним символа

Примеры.

а) Выбрать список деталей, начинающихся с буквы "Б"[1]

Select номер_детали, название, вес from P where название like "Б%"

Результат:	номер_детали	название	вес
	P5	Болт	12
	P6	Блюм	19

II. Использование функций

1. Агрегатные функции.

Примеры.

а) Выдать общее количество поставщиков.

Select count (*) from S

Результат: 5

Подготовьте запрос и проверьте полученный результат.

б) Выдать общее количество поставщиков, поставляющих в настоящее время детали.

Select count (distinct номер_поставщика) from SPJ

Результат: 4

Подготовьте запрос и проверьте полученный результат.

в) Выдать количество поставок для детали P2.

Select count (*) from SPJ where номер_детали='P2'

Результат: 5

Подготовьте запрос и проверьте полученный результат.

г) Выдать общее количество поставляемых деталей 'P2'.

Select sum (количество) from SPJ where номер_детали='P2'

Результат: 1000

Подготовьте запрос и проверьте полученный результат.

[1] Примечание. Корректно работает только при задании кодировки по умолчанию. Задается в разделе MYSQLD default_character_set=win1251

д) Выдать средний, минимальный и максимальный объем поставок для поставщика S1 с соответствующим заголовком.

Select avg(количество) average, min(количество) minimum,

max(количество) maximum from SPJ where номер_поставщика='S1'

Результат:	average	minimum	maximum
	216.6	100	400

Подготовьте запрос и проверьте полученный результат.

2. Ниже привиден перечень всех функций, используемых в операторе Select

Функции

select_expression может содержать следующие функции и операторы:

+ - * /	Арифметические действия.
%	Остаток от деления (как в C)
| &	Битовые функции (используется 48 бит).
- C	Мена знака числа.
( )	Скобки.
BETWEEN(A, B, C)	(A >= B) AND (A <= C).
BIT_COUNT()	Количество бит.
ELT(N, a, b, c, d)	Возвращает a, если N == 1, b, если N == 2 и т. д. a, b, c, d строки. **ПРИМЕР:** ELT(3, "First", "Second", "Third", "Fourth") вернет "Third".
FIELD(Z, a, b, c)	Возвращает a, если Z == a, b, если Z == b и т. д. a, b, c, d строки. **ПРИМЕР:** FIELD("Second", "First", "Second", "Third", "Fourth") вернет "Second".
IF(A, B, C)	Если A истина (!= 0 and != NULL), то вернет B, иначе вернет C.
IFNULL(A, B)	Если A не null, вернет A, иначе вернет B.
ISNULL(A)	Вернет 1, если A == NULL, иначе вернет 0.

	Эквивалент ('A == NULL').
NOT !	NOT, вернет TRUE (1) или FALSE (0).
OR, AND	Вернет TRUE (1) или FALSE (0).
SIGN()	Вернет -1, 0 или 1 (знак аргумента).
SUM()	Сумма столбца.
= <> <= < >= >	Вернет TRUE (1) или FALSE (0).
expr LIKE expr	Вернет TRUE (1) или FALSE (0).
expr NOT LIKE expr	Вернет TRUE (1) или FALSE (0).
expr REGEXP expr	Проверяет строку на соответствие регулярному выражению expr.

select_expression может также содержать один или большее количество следующих математических функций.

ABS()	Абсолютное значение (модуль числа).
CEILING()	()
EXP()	Экспонента.
FORMAT(nr, NUM)	Форматирует число в формат '#, ###, ###.##' с NUM десятичных цифр.
LOG()	Логарифм.
LOG10()	Логарифм по основанию 10.
MIN(), MAX()	Минимум или максимум соответственно. Должна иметь при вызове два или более аргументов, иначе рассматривается как групповая функция.
MOD()	Остаток от деления (аналог %).
POW()	Степень.
ROUND()	Округление до ближайшего целого числа.
RAND([integer_expr])	Случайное число типа float, 0 <= x <= 1.0, используется integer_expr как значение для запуска генератора.
SQRT()	Квадратный корень.

select_expression может также содержать одну или больше следующих строковых функций.

CONCAT()	Объединение строк.

INTERVAL(A, a, b, c, d)	Возвращает 1, если A == a, 2, если A == b... Если совпадений нет, вернет 0. A, a, b, c, d... строки.
INSERT(org, strt, len, new)	Заменяет подстроку org[strt...len(gth)] на new. Первая позиция строки=1.
LCASE(A)	Приводит A к нижнему регистру.
LEFT()	Возвращает строку символов, отсчитывая слева.
LENGTH()	Длина строки.
LOCATE(A, B)	Позиция подстроки B в строке A.
LOCATE(A, B, C)	Позиция подстроки B в строке A, начиная с позиции C.
LTRIM(str)	Удаляет все начальные пробелы из строки str.
REPLACE(A, B, C)	Заменяет все подстроки B в строке A на подстроку C.
RIGHT()	Get string counting from right.
RTRIM(str)	Удаляет хвостовые пробелы из строки str.
STRCMP()	Возвращает 0, если строки одинаковые.
SUBSTRING(A, B, C)	Возвращает подстроку из A, с позиции B до позиции C.
UCASE(A)	Переводит A в верхний регистр.

Еще несколько просто полезных функций, которые тоже можно применить в select_expression.

CURDATE()	Текущая дата.
DATABASE()	Имя текущей базы данных из которой выполняется выбор.
FROM_DAYS()	Меняет день на DATE.
NOW()	Текущее время в форматах YYYYMMDDHHMMSS или "YYYY-MM-DD HH:MM:SS". Формат зависит от того в каком контексте используется NOW(): числовом или строковом.
PASSWORD()	Шифрует строку.
PERIOD_ADD(P:N)	Добавить N месяцев к периоду P (в формате YYMM).
PERIOD_DIFF(A, B)	Возвращает месяцы между A и B. Обратите внимание, что PERIOD_DIFF работает только с датами в форме YYMM или YYYMM.

TO_DAYS()	Меняет DATE (YYMMDD) на номер дня.
UNIX_TIMESTAMP([date])	Возвращает метку времени unix, если вызвана без date (секунды, начиная с GMT 1970.01.01 00:00:00). При вызове со столбцом TIMESTAMP вернет TIMESTAMP. date может быть также строкой DATE, DATETIME или числом в формате YYMMDD (или YYYMMDD).
USER()	Возвращает логин текущего пользователя.
WEEKDAY()	Возвращает день недели (0 = понедельник, 1 = вторник, ...).

Групповые функции в операторе ***select*:**

Следующие функции могут быть использованы в предложении GROUP:

AVG()	Среднее для группы GROUP.
SUM()	Сумма элементов GROUP.
COUNT()	Число элементов в GROUP.
MIN()	Минимальный элемент в GROUP.
MAX()	Максимальный элемент в GROUP.

Задание:

1. Подготовить 3 запроса с использованием различных функций работа с полем дата, со строковыми данными (в том числе групповых).

2. Подготовить и выполнить средствами СУБД MySQL 4 запроса по выборке информации из таблиц базы данных с использованием агрегатных функций..

4. Подготовить и выполнить средствами СУБД MySQL 2 запроса по модификации информации (вставка, удаление, замещение) из таблиц базы данных для решения нижеприведенных задач. При этом в тех заданиях, где речь идет о создании таблиц, предполагается формировании постоянной таблицы базы данных.

Лабораторная работа №3. Представления, курсоры, хранимые процедуры, функции, триггеры

Цель работы: познакомиться с возможностями MySQL по работе с хранимыми процедурами, функциями, триггерами, представлениями.

Представления

Представления (views) можно сравнить с временными таблицами, наполненными динамически формируемым содержимым.. В настоящей реализации есть две возможности создания представлений: с использованием алгоритма временных таблиц MySQL и с созданием самостоятельной таблицы. Нас интересует именно второй способ (первый был реализован, скорее всего, исходя из соображений совместимости и унификации). Такие представления позволяют значительно снизить объём кода, в котором часто повторялись простые объединения таблиц. К ним (после создания) применимы любые запросы, возвращающие результат в виде набора строк. То есть команды SELECT, UPDATE, DELETE, можно применять так же, как и к реальным таблицам. Важно и то, что посредством представлений можно более гибко распоряжаться правами пользователей базы данных, так как в этом случае есть возможность предоставлять доступ на уровне отдельных записей различных таблиц.

Создание представлений

Для создания представлений используется команда CREATE VIEW

Синтаксис команды CREATE VIEW

```
CREATE
   [OR REPLACE]
   [ALGORITHM = {UNDEFINED | MERGE | TEMPTABLE}]
   [DEFINER = { user | CURRENT_USER }]
   [SQL SECURITY { DEFINER | INVOKER }]
   VIEW view_name [(column_list)]
   AS select_statement
   [WITH [CASCADED | LOCAL] CHECK OPTION]
```

Пример создания и работы простейшего представления:

Create View v as Select column 1 from T

Insert into v Values (1)

Select * from v

Результат

```
+--------+
| column1 |
+--------+
```

```
| 1    |
+---------+
1 row in set (0.00 sec)
```

Представление может быть создано на основе различных параметров предложения SELECT, при этом можно ссылаться на другие таблицы и представления. Конструкция может использовать оператор UNION и другие подзапросы.

Синтаксис команды ALTER VIEW

Для внесения изменений в представление используется команда ALTER VIEW

```
ALTER
   [ALGORITHM = {UNDEFINED | MERGE | TEMPTABLE}]
   [DEFINER = { user | CURRENT_USER }]
   [SQL SECURITY { DEFINER | INVOKER }]
   VIEW view_name [(column_list)]
   AS select_statement
   [WITH [CASCADED | LOCAL] CHECK OPTION]
```

Синтаксис команды DROP VIEW

Для удаления представления используется команда DROP VIEW

```
VIEW [IF EXISTS]
   view_name [, view_name] ...
   [RESTRICT | CASCADE]
```

ПРИМЕР

```
mysql> CREATE TABLE t (qty INT, price INT);
mysql> INSERT INTO t VALUES(3, 50);
mysql> CREATE VIEW v AS SELECT qty, price, qty*price AS value FROM t;
mysql> SELECT * FROM v;
+------+-------+-------+
| qty  | price | value |
+------+-------+-------+
|    3 |    50 |   150 |
```

Хранимые процедуры и функции

В СУБД MySQL появилась возможность создания и хранения функций и процедур. Объявление и работа с процедурами и функциями отличаются в следующем:

- в заголовке функции помимо описания формальных параметров обязательно указывается тип возвращаемого ею результата;
- для возврата функцией значения в точку вызова среди ее операторов должен быть хотя бы один, в котором имени функции или переменной Result присваивается значение результата;

- вызов процедуры выполняется отдельным оператором;
- вызов функции может выполняться там, где допускается ставить выражение, в частности, в правой части оператора присваивания.

Пользовательские функции по функциональности похожи на хранимые процедуры. Разница заключается в том, что возможностей у них меньше (в частности, они должны возвращать только одно значение, например, скалярное или табличное), но их удобнее использовать с точки зрения синтаксиса.

Как процедуры, так и функции могут возвращать значения (в виде набора записей). Различие состоит в том, что функция вызывается из запроса, а процедура из отдельной команды.

На настоящий момент реализация хранимых процедур не поддерживает никаких внешних языков, но (по крайней мере, так заявляется) соответствует стандарту SQL:2003, позволяющему применять условные конструкции, итерации и обработку ошибок.

Пример создания хранимой процедуры в MySQL 5:

CREATE PROCEDURE p ()

LANGUAGE SQL

NOT DETERMINISTIC

SQL SECURITY DEFINER

COMMENT 'A Procedure' <--

SELECT CURRENT_DATE, RAND() FROM t

В данном случае мы создали процедуру с именем p, которая возвращает текущую дату и псевдослучайное число из таблицы t. Пример ее вызова и возвращаемого результата:

```
mysql> call p2()
+--------------+----------------+
| CURRENT_DATE | RAND()         |
+--------------+----------------+
| 2005-06-27   | 0.7822275075896 |
+--------------+----------------+
1 row in set (0.26 sec)
Query OK, 0 rows affected (0.26 sec)
```

Чуть более сложный пример создания и использования функции:

CREATE FUNCTION factorial (n DECIMAL(3,0))

```
RETURNS DECIMAL(20,0)
DETERMINISTIC
BEGIN
DECLARE factorial DECIMAL(20,0) DEFAULT 1;
DECLARE counter DECIMAL(3,0);
SET counter = n;
factorial_loop: REPEAT
SET factorial = factorial * counter;
SET counter = counter - 1;
UNTIL counter = 1
END REPEAT;
RETURN factorial;
END
```

В приложении:

```
INSERT INTO t VALUES (factorial(pi))
SELECT s1, factorial (s1) FROM t
UPDATE t SET s1 = factorial(s1)
WHERE factorial(s1) < 5
```

Разумеется эффективность применения хранимых процедур существенно возрастает при вызове их с параметрами (аргументами). Ниже дан пример процедуры с обработкой переданных ей параметров:

```
CREATE PROCEDURE p1 (IN parameter1 INT)
BEGIN
DECLARE variable1 INT;
SET variable1 = parameter1 + 1;
IF variable1 = 0 THEN
INSERT INTO t VALUES (17);
END IF;
IF parameter1 = 0 THEN
```

```
UPDATE t SET s1 = s1 + 1; <--
ELSE
UPDATE t SET s1 = s1 + 2;
END IF;
END;
```

Вызов процедуры теперь будет таким:

```
mysql> CALL p2(0) // Query OK, 2 rows affected (0.28 sec)
```

и в результате запроса мы получим:

```
mysql> SELECT * FROM t
+----+
| s1 |
+----+
| 6  |
| 6  |
+------+
2 rows in set (0.01 sec)
```

Кроме условных, возможны и любые циклические конструкции:

```
CREATE PROCEDURE p3 ()
BEGIN
DECLARE v INT;
SET v = 0;
WHILE v < 5 DO
INSERT INTO t VALUES (v);
SET v = v + 1;
END WHILE;
END;
```

Вызов процедуры:

```
mysql> CALL p3()
```

```
+------+
| s1   |
+------+
……….


|   0 |
|   1 |
|   2 |
|   3 |
|   4 |
+------+
Query OK, 1 row affected (0.00 sec)
```

Также применимы итерации, переходы, словом, всё, что предполагает стандарт.

Внутри функций и хранимых процедур осуществлена реализация курсоров, но, к сожалению, она пока ограничена (ASESITIVE, READ ONLY и NONSCROLL):

```
CREATE PROCEDURE p25 (OUT return_val INT)
BEGIN
DECLARE a,b INT;
DECLARE cur_1 CURSOR FOR SELECT s1 FROM t;
DECLARE CONTINUE HANDLER FOR NOT FOUND
SET b = 1;
OPEN cur_1;
REPEAT
FETCH cur_1 INTO a;
UNTIL b = 1
END REPEAT;
CLOSE cur_1;
SET return_val = a;
END;
```

Создание процедур и функций

CREATE

[DEFINER = { *user* | CURRENT_USER }]

PROCEDURE *sp_name* ([*proc_parameter*[,...]])

[*characteristic* ...] *routine_body*

CREATE

[DEFINER = { *user* | CURRENT_USER }]

FUNCTION *sp_name* ([*func_parameter*[,...]])

RETURNS *type*

[*characteristic* ...] *routine_body*

proc_parameter:

[IN | OUT | INOUT] *param_name type*

func_parameter:

param_name type

type:

Any valid MySQL data type

characteristic:

LANGUAGE SQL

| [NOT] DETERMINISTIC

| { CONTAINS SQL | NO SQL | READS SQL DATA | MODIFIES SQL DATA }

| SQL SECURITY { DEFINER | INVOKER }

| COMMENT '*string*'

routine_body:

Внесение изменений

ALTER {PROCEDURE | FUNCTION} *sp_name* [*characteristic* ...]

characteristic:
  { CONTAINS SQL | NO SQL | READS SQL DATA | MODIFIES SQL DATA }
 | SQL SECURITY { DEFINER | INVOKER }
 | COMMENT '*string*'

Удаление процедур и функций

DROP {PROCEDURE | FUNCTION} [IF EXISTS] *sp_name*

Вызов процедур и функций

CALL *sp_name*([*parameter*[,...]])
CALL *sp_name*[()]

Оператор CALL позволяет вызвать ранее определенную процедуру.

Пример1

```
CREATE PROCEDURE p1 (OUT ver_param VARCHAR(25), INOUT incr_param INT)
BEGIN
 # Set value of OUT parameter
 SELECT VERSION() INTO ver_param;
 # Increment value of INOUT parameter
 SET incr_param = incr_param + 1;
END;
```

Перед вызовом процедуры инициализируйте переменную указанные в параметрах INOUT . После вызова процедуры значения будут установлены или изменены.

```
mysql> SET @increment = 10;
mysql> CALL p(@version, @increment);
mysql> SELECT @version, @increment;
+----------------+------------+
| @version       | @increment |
+----------------+------------+
| 5.1.12-beta-log | 11         |
```

Пример2

```
CREATE PROCEDURE `p2`(IN param1 CHAR(2) )
   NOT DETERMINISTIC
   SQL SECURITY DEFINER
   COMMENT ''
BEGIN
select * from s where snum=param1;
END;
```

Вызов процедуры
call p2 ('S1')

Пример3

```
CREATE PROCEDURE `My_proc2`(IN param1 CHAR(2) )

BEGIN                          /* start of block */
 DECLARE variable1 CHAR(10);           /* variables */
 IF param1 = 17 THEN                /* start of IF */
  SET variable1 = 'birds';             /* assignment */
 ELSE
  SET variable1 = 'beasts';            /* assignment */
 END IF;                         /* end of IF */
select variable1;   /* statement */
END
```

Вызов процедуры
call p3 (10)

Триггеры

Три́ггер (англ. trigger) — это хранимая процедура особого типа, которую пользователь не вызывает непосредственно, а исполнение которой обусловлено наступлением определенного события (действием) — по сути добавлением INSERT или удалением DELETE строки в заданной таблице, или модификаци UPDATE данных в определенном столбце заданной таблицы реляционной базы данных. Триггеры применяются для обеспечения целостности данных и реализации сложной бизнес-логики. Триггер запускается сервером автоматически при попытке изменения данных в таблице, с которой он связан. Все производимые им модификации данных рассматриваются как выполняемые в транзакции, в которой выполнено действие, вызвавшее срабатывание триггера. Соответственно, в случае обнаружения ошибки или нарушения целостности данных может произойти откат этой транзакции. Момент запуска триггера определяется с помощью ключевых слов BEFORE (триггер запускается до выполнения связанного с ним события; например, до добавления записи) или AFTER (после события). В случае, если триггер вызывается до события, он может внести изменения в модифицируемую событием запись (конечно, при условии, что событие — не удаление записи). Некоторые СУБД накладывают ограничения на операторы, которые могут быть использованы в триггере (например, может быть запрещено вносить изменения в таблицу, на которой «висит» триггер, и т. п.)

Кроме того, триггеры могут быть привязаны не к таблице, а к представлению (VIEW). В этом случае с их помощью реализуется механизм «обновляемого представления». В этом случае ключевые слова BEFORE и AFTER влияют лишь на последовательность вызова триггеров, так как собственно событие (удаление, вставка или обновление) не происходит.

```
CREATE
    [DEFINER = { user | CURRENT_USER }]
    TRIGGER trigger_name trigger_time trigger_event
    ON tbl_name FOR EACH ROW trigger_stmt
```

Пример создания и работы триггера:

```
CREATE TABLE t22 (s1 INTEGER)

CREATE TRIGGER t22_bi
BEFORE INSERT ON t22
FOR EACH ROW
BEGIN
SET @x = ‘Trigger was activated!’;
SET NEW.s1 = 55;
END;
```

После этого при выполнении запросов получим:

```
mysql> INSERT INTO t22 VALUES (1)

mysql> SELECT @x, t22.* FROM t22 // вызывается триггер

+----------------------+------+

| @x                   | s1   |

+----------------------+------+

| Trigger was activated! | 55   |

+----------------------+------+

1 row in set (0.00 sec)
```

Словарь данных

Иметь доступ к значениям метаданных – совершенно необходимое требование к современной СУБД. Ранее такая возможность в MySQL достигалась различными SHOW-командами, но такой подход имеет очевидные недостатки. Эти команды нельзя использовать в простых запросах с соединениями, и, что существенно, они не соответствовали стандартам, будучи специфичными для MySQL.

В новой версии СУБД появилась новая служебная база данных – INFORMATION_SCHEMA. Её наличие продиктовано тем же стандартом SQL:2003, и именно она решает задачу реализации словаря данных (data dictionary). INFORMATION_SCHEMA содержит таблицы, описывающие состояние и параметры сервера, в том числе определения и сущности таблиц. Это виртуальная база данных – физически (в виде файлов на диске) она не существует, вся информация динамически предоставляется сервером. Пример использования этой таблицы:

```
mysql> SELECT table_name, table_type, engine

   -> FROM INFORMATION_SCHEMA.tables

   -> WHERE table_schema = 'tp'

   -> ORDER BY table_type ASC, table_name DESC;

+------------+------------+--------+

| table_name | table_type | engine |

+------------+------------+--------+

| t2         | BASE TABLE | MyISAM |

| t1         | BASE TABLE | InnoDB |

| v1         | VIEW       | NULL   |

+------------+------------+--------+
```

Другой пример работы со словарём данных – просмотр привелегий:

```
mysql> SELECT * FROM
    -> INFORMATION_SCHEMA.COLUMN_PRIVILEGES\G
*************************** 1. row ***************************
GRANTEE: 'peter'@'%'
TABLE_CATALOG: NULL
TABLE_SCHEMA: tp
TABLE_NAME: t1
COLUMN_NAME: col1
PRIVILEGE_TYPE: UPDATE
IS_GRANTABLE: NO
*************************** 2. row ***************************
GRANTEE: 'trudy'@'%'
TABLE_CATALOG: NULL
TABLE_SCHEMA: tp
TABLE_NAME: t2
COLUMN_NAME: col1
PRIVILEGE_TYPE: SELECT
IS_GRANTABLE: YES
```

Объявление переменных

Объявление. DECLARE Local Variables

Следующая команда позволяет объявлять локальные переменные, содержит возможность задания значения по умолчанию. Переменная может быть объявлена как выражения, не обязательна константа. Если значение по умолчанию не определено то равно NULL.
DECLARE *var_name*[,...] *type* [DEFAULT *value*]

Присваивание Variable SET Statement

SET *var_name* = *expr* [, *var_name* = *expr*] ...

SELECT ... INTO Statement

Оператор SELECT может перенаправить результат в переменные. Таким образом может быть преобразована только одна строка.

ПРИМЕР

SELECT *col_name*[,...] INTO *var_name*[,...] *table_expr*
SELECT id,data INTO x,y FROM test.t1 LIMIT 1;

Условия и ограничения

Объявление условий

DECLARE *condition_name* CONDITION FOR *condition_value*

condition_value:
  SQLSTATE [VALUE] *sqlstate_value*
 | *mysql_error_code*

Объявление ограничений

DECLARE *handler_type* HANDLER FOR *condition_value*[,...] *statement*

handler_type:
  CONTINUE
 | EXIT
 | UNDO

condition_value:
  SQLSTATE [VALUE] *sqlstate_value*
 | *condition_name*
 | SQLWARNING
 | NOT FOUND
 | SQLEXCEPTION
 | *mysql_error_code*

Пример

mysql> **CREATE TABLE test.t (s1 int,primary key (s1));**
Query OK, 0 rows affected (0.00 sec)

```
mysql> delimiter //

mysql> CREATE PROCEDURE handlerdemo ()
    -> BEGIN
    ->   DECLARE CONTINUE HANDLER FOR SQLSTATE '23000' SET @x2 = 1;
    ->   SET @x = 1;
    ->   INSERT INTO test.t VALUES (1);
    ->   SET @x = 2;
    ->   INSERT INTO test.t VALUES (1);
    ->   SET @x = 3;
    -> END;
    -> //
Query OK, 0 rows affected (0.00 sec)

mysql> CALL handlerdemo()//
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT @x//
   +------+
   | @x   |
   +------+
   | 3    |
   +------+
   1 row in set (0.00 sec)
```

Если вы хотите игнорировать условие вы должны сгенерировать ограничение и ассоциировать его с пустым блоком .

```
DECLARE CONTINUE HANDLER FOR SQLWARNING BEGIN END;
```

Пример

```
CREATE PROCEDURE p ()
BEGIN
 DECLARE i INT DEFAULT 3;
 retry:
   REPEAT
     BEGIN
       DECLARE CONTINUE HANDLER FOR SQLWARNING
         BEGIN
           ITERATE retry;  # illegal
         END;
     END;
     IF i < 0 THEN
       LEAVE retry;       # legal
     END IF;
     SET i = i - 1;
   UNTIL FALSE END REPEAT;
END;
```

Курсоры

Курсор — в некоторых реализациях языка программирования SQL (Oracle, Microsoft SQL Server) — получаемый при выполнении запроса результирующий набор и связанный с ним указатель текущей записи.

Курсор может возвращать одну строку, несколько строк или ни одной строки. Для запросов, возвращающих более одной строки, можно использовать только явный курсор. Для повторного создания результирующего набора для других значений параметров курсор следует закрыть, а затем повторно открыть.

Курсор может быть объявлен в секциях объявлений любого блока PL/SQL, подпрограммы или пакета.

Операторы управления явным курсором

- Оператор CURSOR выполняет объявление явного курсора.
- Оператор OPEN открывает курсор, создавая новый результирующий набор на базе указанного запроса.
- Оператор FETCH выполняет последовательное извлечение строк из результирующего набора от начала до конца.
- Оператор CLOSE закрывает курсор и освобождает занимаемые им ресурсы

Курсоры поддерживают хранимые процедуры и функции. Сейчас курсоры имеют три свойства:

- Asensitive: The server may or may not make a copy of its result table
- Read only: Not updatable
- Non-scrollable: Can be traversed only in one direction and cannot skip rows

Курсоры должны быть объявлены перед объявлением ограничений. Переменные и условия должны быть объявлены перед курсором.

Объявление курсоров

Оператор объявления курсора. М программе можно объявлять несколько курсоров, каждый курсор в блоке должен иметь уникальное имя.

DECLARE *cursor_name* CURSOR FOR *select_statement*

Условие открытия Cursor OPEN Statement

Оператор открывает ранее объявленный курсор.

OPEN *cursor_name*

Выполнение курсора Cursor FETCH Statement

FETCH *cursor_name* INTO *var_name* [, *var_name*] ...

Условия закрытия Cursor CLOSE Statement

CLOSE *cursor_name*

Пример

```
CREATE PROCEDURE curdemo()
BEGIN
  DECLARE done INT DEFAULT 0;
  DECLARE a CHAR(16);
  DECLARE b,c INT;
  DECLARE cur1 CURSOR FOR SELECT id,data FROM test.t1;
  DECLARE cur2 CURSOR FOR SELECT i FROM test.t2;
  DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

  OPEN cur1;
  OPEN cur2;

  REPEAT
    FETCH cur1 INTO a, b;
    FETCH cur2 INTO c;
    IF NOT done THEN
      IF b < c THEN
        INSERT INTO test.t3 VALUES (a,b);
      ELSE
        INSERT INTO test.t3 VALUES (a,c);
      END IF;
    END IF;
  UNTIL done END REPEAT;

  CLOSE cur1;
  CLOSE cur2;
END
```

Лабораторная работа 4

Задание

1 Корреляция и регрессионный анализ
2 Парадигма Map Reduce.
3 Основные понятия теории нейронных сетей.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Назначение дисциплины.
2. Основные понятия
3. Роль аналитика данных (Data Scientist).
4. Ключевые компетенции аналитика.
5. Отличия BI от Data Science.
6. Общая схема анализа.
7. Извлечение и визуализация данных.
8. Этапы моделирования.
9. Процесс построения моделей.
10. Формы представления данных, типы и виды данных.
11. Представления наборов данных.
12. . Коэффициент корреляции.
13. Графическое представление.
14. Постановка задачи регрессионного анализа.
15. Линейная регрессия.
16. Метод наименьших квадратов.
17. Их роль в аналитике больших данных.
18. Ассоциативные правила.
19. Постановка задачи классификации.
20. Постановка задачи кластеризации.
21. Задача построения ассоциативных правил.
22. Подготовка данных к анализу.
23. Методика извлечения знаний.
24. Data Mining.
25. Мультидисциплинарный характер Data Mining.
26. Причины распространения KDD и Data Mining.
27. Актуальность технологий Data Mining как средств обработки больших объемов информации..
28. Реализация Hadoop.
29. Парадигма Map Reduce.
30. Роль Map Reduce в аналитике больших данных.
31. Аналитические платформы: классификация и особенности применения.
32. Языки визуального моделирования.
33. Начало работы.
34. Понятие сценария и узла обработки.
35. Консолидация данных.
36. Трансформация данных.
37. Визуализация данных.
38. Понимание данных.
39. Методы предварительной подготовки данных.
40. Инструменты и методы визуализации данных
41. Регуляризация.
42. Нейронные сети.
43. Машина опорных векторов.
44. Разбор алгоритма нейронных сетей.
45. Разбор алгоритма SVM
46. Основные парадигмы нейронных сетей.
47. Многослойный персептрон: класс решаемых задач, архитектура.
48. Причины популярности и условия применимости.
49. Структура дерева решений.
50. Выбор атрибута разбиения в узле.
51. Алгоритм ID3, критерий выбора атрибута разбиения ID3, пример работы алгоритма.

52. Проблема переобучения,.
53. Неизвестные значения атрибутов, алгоритм C4.5.

ТЕСТЫ

Основные понятия и технологии работы с реляционными базами данных/Понятие систем управления базами данных

1. **СУБД работающие с какой моделью данных получили самое широкое распространение:**

- ○ Списки
- ● Реляционной
- ○ Сетевой
- ○ Иерархической

2. **Основное назначение менеджера транзакций:**

- ○ Управление оперативной памятью
- ○ Найти лучший способ выполнения требуемой операции и дать соответствующие команды менеджеру памяти
- ○ Контролирует расположение файлов на диске
- ● Гарантировать правильное выполнение всех транзакций

3. **Основные требования, предъявляемые к выполнению транзакций (отметьте четыре варианта):**

- ● **25%** Атомарность
- ● **25%** Непротиворечивость
- ● **25%** Изоляция
- ● **25%** Долговременность
- ○ **-100%** Запрет на взаимодействие с запросами

4. **Подсистема обработки применяется для:**

- ○ Выполнения функции посредника между подсистемой средств проектирования и обработки и данными
- ● Обработки компонентов приложений, созданных с помощью средств проектирования
- ○ Упрощения проектирования и реализации баз данных и их приложений

5. **Для чего служат метаданные?**

- ○ Управление основной памятью
- ● Хранение информации о структуре данных
- ○ Обработки сбоев
- ○ Упрощение проектирования

6. **По степени универсальности различают СУБД (отметьте два варианта):**

- -100% Сетевые
- 50% Общего назначения
- -100% Реляционные
- -100% Иерархические
- 50% Специального назначения

7. Какой язык используется для работы в современных СУБД?

- DML
- SDL
- SQL

8. Какие СУБД относятся к категории персональных? (отметьте два варианта)

- 50% FoxBase
- -100% Sybase
- 50% FoxPro
- -100% Inqres

9. СУБД это:

- Часть реального мира, подлежащая изучению с целью создания базы данных для автоматизации процесса управления
- Любой конкретный или абстрактный объект в рассматриваемой предметной области.
- Это свойство сущности в предметной области
- Это совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями

10. Механизм запросов используется для:

- Добавление записей
- Удаление записей
- Написания сторонних приложений
- Обновления записей

Основные понятия и технологии работы с реляционными базами данных/Модели ранних СУБД

11. Достоинства ранних СУБД (отметьте три варианта):

- 33% Развитые средства управления данными во внешней памяти на низком уровне
- 33% Возможность построения вручную эффективных прикладных систем
- 34% Возможность экономии памяти за счет разделения подобъектов (в сетевых системах)
- -100% Прикладные системы зависят от этой организации

12. Недостатки ранних СУБД (отметьте три варианта):

- **33%** Слишком сложно пользоваться
- **33%** Их логика перегружена деталями организации доступа к БД
- **34%** Фактически необходимы знания о физической организации
- **-100%** Возможность построения вручную эффективных прикладных систем

13. Какая модель данных изображена на рисунке:

- Сетевая модель
- Иерархическая модель
- Реляционная модель данных
- Информационно-логическая модель данных

14. Разница между иерархической моделью данных и сетевой состоит:

- В том, что в иерархических структурах запись-потомок должна иметь в точности одного предка, а в сетевой структуре данных у потомка может иметься любое число предков
- В том, что в сетевых структурах запись-потомок должна иметь в точности одного предка, а в иерархической структуре данных у потомка может иметься любое число предков

15. Какая модель данных является самой распространенной в настоящее время?

- Сетевая модель
- Иерархическая модель
- Реляционная модель данных
- Информационно-логическая модель данных

16. Почему СУБД компании Microsoft получили наибольшее распространение?

- Они имеют большие возможности интеграции, совместной работы и использования данных, так как данные пакеты являются продуктами одного производителя, а также используют сходные технологии обмена данными
- Из-за небольшой стоимости
- Из-за большой популярности компании Microsoft

17. Укажите СУБД компании Microsoft (отметьте два варианта):

- **-100%** Oracle
- **50%** SQL Server
- **50%** Visual FoxPro
- **-100%** Lotus Approach
- **-100%** dBase

18. Технология «Клиент-сервер» изображена на рисунке?

Функции:
интерфейс пользователя,
логика обработки,
управление данными

○

Функции:
интерфейс пользователя,
логика обработки

●

19. Администрирование базы данных это:

- ● Это функция управления базой данных
- ○ Это лицо, отвечающее за выработку требований к базе данных, её проектирование, реализацию, эффективное использование и сопровождение

Основные понятия и технологии работы с реляционными базами данных/Реляционная модель данных

20. Какой из модели данных соответствует данное утверждение: “Данные в БД представляют собой набор отношений”?

- ○ Сетевая модель данных
- ● Реляционная модель данных
- ○ Иерархическая модель данных
- ○ Не соответствует ни одной из моделей данных

21. Что характерно для реляционной модели данных? (отметьте три варианта)

- **33%** Использования понятия отношений и таблиц
- **-100%** Древовидная структура
- **33%** Модель является логической
- **-100%** Модель является физической
- **-100%** Характеризуется такими понятиями как уровни, узлы и связи
- **34%** Использование теории нормализации

22. Какие из перечисленных объектов можно отнести к объектам реляционной модели данных? (отметьте три варианта)

- **33%** Кортеж
- **-100%** Узел
- **-100%** Уровень
- **33%** Таблица
- **34%** Отношение
- **-100%** Предок

23. Что такое кортеж? Выберите верное утверждение.

- Это элемент отношения, строка таблицы; упорядоченный набор из N элементов
- Поле, элемент данных
- Отношение между таблицами
- Ни одно из утверждений не верно

24. Какие утверждения верны для таблицы в реляционной БД? (отметьте три варианта)

- **-100%** Могут присутствовать одинаковые строки
- **33%** Все ячейки в столбце однородны
- **-100%** Ячейки в столбце могут быть неоднородными
- **33%** Порядок следования строк в таблице произволен
- **34%** Каждый элемент таблицы (строка) – один элемент данных
- **-100%** Порядок следования строк строго определен
- **-100%** Ни одно из утверждений не верно

25. Почему следует выполнять условие целостности данных?

- Для удобства
- Обязательное требование, необходимое для корректного функционирования БД
- Условие целостности не обязательно соблюдать

26. Что из себя представляет целостность по существованию?

- Целостность по существованию определяется понятием внешнего ключа, и определяет набор правил для поддержания целостности и связей между таблицами

- ○ Между таблицами должны существовать связи
- ● Потенциальный ключ отношения не может принимать значение NULL
- ○ Нет верного утверждения

27. Что такое первичный ключ?

- ● Потенциальный ключ, который берется в качестве основного ключа, и который однозначно и уникально характеризует запись в таблице
- ○ Первое поле в таблице
- ○ Поле, которое ссылается на ключ внешней таблицы
- ○ Верного утверждения нет

28. Почему рекомендуется использовать суррогатные ключи, а не естественные? (отметьте два варианта)

- ● **50%** У естественных ключей низкая эффективность
- ○ **-100%** Суррогатные ключи повторяются
- ● **50%** Легче проводить каскадное обновление таблиц
- ○ **-100%** Суррогатные ключи не рекомендуется использовать

29. Благодаря чему устанавливается связь между отношениями?

- ○ Ссылкам
- ○ Альтернативным ключам
- ○ Первичному ключу
- ● Внешнему ключу

30. Использования внешних ключей позволяет (отметьте два варианта):

- ● **50%** Организовать связи между отношениями
- ● **50%** Избежать дублирования данных
- ○ **-100%** Упростить структуру БД
- ○ **-100%** Ни одно из утверждений не верно

31. Пользователю необходимо удалить запись из родительской таблицы, причем у нее имеются подчиненная таблица, в которой используется эта запись. Чтобы удалить запись в родительской таблице, при этом не удаляя записи с этим внешним ключом в подчиненной, какие правила можно установить? (отметьте два варианта)

- ○ **-100%** Restrict
- ○ **-100%** Cascade
- ● **50%** Set Null
- ● **50%** Set Default

32. Чтобы выполнялось обновление, как в родительской таблице, так и в подчиненных, необходимо установить правило:

- ○ Restrict

- Cascade
- Set Null
- Set Default

Основные понятия и технологии работы с реляционными базами данных/Наиболее распространенные виды моделей данных

33. Иерархическая модель данных — логическая модель данных в виде:

- Двоичное дерево поиска
- Очередь с приоритетом
- Древовидной структуры
- Ассоциативного массива

34. Назовите достоинства иерархической модели данных (отметьте два варианта):

- **50%**Эффективное использование памяти ЭВМ
- **-100%**Громоздкость для обработки информации с достаточно сложными логическими связями
- **-100%**Сложность понимания для обычного пользователя
- **50%**Неплохие показатели времени выполнения основных операций над данными
- **-100%**Нет правильных ответов

35. Назовите недостатки иерархической модели данных (отметьте два варианта):

- **-100%**Нет возможности работать по сети
- **50%**Сложность понимания для обычного пользователя
- **-100%**Эффективное использование памяти ЭВМ
- **50%**Громоздкость для обработки информации с достаточно сложными логическими связями
- **-100%**Нет правильных вариантов ответов

36. Сетевая модель данных — логическая модель данных, являющаяся расширением иерархического подхода, строгая математическая теория, описывающая структурный аспект, аспект целостности и аспект обработки данных в сетевых базах данных. Разница между иерархической моделью данных и сетевой состоит в том, что в иерархических структурах запись-потомок должна иметь в точности одного предка, а в сетевой структуре данных у потомка может иметься:

- Любое число предков
- Один предок
- Один потомок
- Нет правильного ответа

37. Назовите достоинства сетевой модели данных:

- Сложность понимания для обычного пользователя

- Эффективное использование памяти ЭВМ
- Высокая сложность и жесткость схемы БД, построенной на ее основе
- Возможность эффективной реализации по показателям затрат памяти и оперативности

38. Назовите недостатки сетевой модели данных:

- Нет возможности работать по сети
- Неплохие показатели времени выполнения основных операций над данными
- Высокая сложность и жесткость схемы БД, построенной на ее основе
- Эффективное использование памяти ЭВМ

39. В реляционной модели данных объекты и взаимосвязи между ними представлены в виде:

- Столбцов
- Графов
- Таблиц
- Деревьев

40. Назовите недостатки реляционной модели данных (отметьте три варианта):

- **33%** По сравнению с иерархической и сетевой моделями реляционная модель имеет более низкую скорость доступа и требует большего объема внешней памяти. В настоящее время этот фактор не является критическим вследствие многократно возросшего быстродействия компьютеров и такого же роста объема дисковой памяти
- **33%** Часто в результате логического проектирования появляется очень много таблиц, что затрудняет понимание структуры данных
- **34%** Далеко не всегда предметную область можно представить в виде совокупности таблиц. Так, в системах автоматизации проектирования и автоматизированной разработки программного обеспечения требуются гораздо более сложные структуры данных
- **-100%** Сложность понимания для обычного пользователя
- **-100%** При проектировании реляционных баз данных применяются строгие правила, базирующиеся на математическом аппарате
- **-100%** Нет правильных вариантов

41. Назовите достоинства реляционной модели данных (отметьте четыре варианта):

- **-100%** Нет правильных вариантов ответа
- **-100%** Нет возможности работать по сети
- **-100%** Сложность понимания для обычного пользователя
- **25%** Одним из важных достоинств реляционного подхода является его простота и доступность для понимания конечным пользователем. Единственной информационной конструкцией является таблица
- **25%** При проектировании реляционных баз данных применяются строгие правила, базирующиеся на математическом аппарате

- **25%** Реляционная модель обеспечивает полную независимость данных. При изменении структуры реляционной базы данных изменения, которые требуется произвести в прикладных программах, как правило, минимальны
- **25%** Манипулирование данными на уровне языка СУБД производится ненавигационно, поэтому для построения запросов и написания прикладных программ нет необходимости знания конкретной организации базы данных во внешней памяти. Конечно, при исполнении запросов на физическом уровне выполняется навигация по записям таблиц, однако эти действия производятся процедурами самой СУБД

42. Для преодоления недостатков, присущих реляционной модели, в настоящее время развивается:

- Сетевая модель
- Иерархическая модель
- Постреляционная модель
- Нет правильных вариантов

Основные понятия и технологии работы с реляционными базами данных/Математический аппарат реляционных БД

43. Операции реляционной алгебры делятся на (отметьте два варианта):

- **50%** Основные
- **-100%** Логические
- **50%** Дополнительные
- **-100%** Арифметические

44. В реляционной алгебре операнды и результаты всех операций являются:

- Отношениями
- Выражениями
- Аргументами

45. В контексте баз данных реляционное исчисление существует в двух формах:

- **50%** В форме реляционного исчисления кортежей
- **50%** В форме реляционного исчисления доменов
- **-100%** В форме реляционного исчисления выражений
- **-100%** В форме реляционного исчисления атрибутов

46. Реляционное исчисление кортежей состоит в отыскании таких кортежей, для которых:

- Предикат является ложным
- Предикат неопределен
- Предикат является истинным

47. Все запросы, которые можно сформулировать с помощью реляционной алгебры:

- Нельзя сформулировать с помощью реляционного исчисления и наоборот

- Можно сформулировать с помощью реляционного исчисления и наоборот
- Не всегда можно сформулировать с помощью реляционного исчисления и наоборот

48. Реляционная алгебра в отличии от реляционного исчисления:

- Задает порядок операций
- Оставляет компилятору определять наиболее эффективный порядок вычисления

49. К реляционным операторам относятся (отметьте два варианта):

- **-100%**And
- **50%**= (Равно)
- **-100%**Or
- **50%**< > (Не равно)

50. Три основные команды DML:

- INSERT, UPDATE, DELETE
- UNION, EXISTS, CREAT
- INSERT, CREAT, EXISTS

51. В каком из вариантов правильно записан формат команды Insert:

- Insert Into имя_таблицы [(список_имен)]
  Values (список_значений);
- Insert Into имя_таблицы [(список_имен)]
  Set поле = значение, [поле = значение], …
  Where условие;
- Insert Into имя_таблицы [(список_имен)]
  Where условие;

52. Команды INSERT, UPDATE, DELETE называются в SQL:

- Командами обновления данных
- Командами выборки данных
- Командами объединения таблиц
- Командами управления БД

Работа с современными промышленными СУБД/Языки управления базами данных

53. К языку манипулирования данными не относится команда:

- Вставить
- Создать
- Обновить
- Удалить

54. Семейство компьютерных языков, используемых в компьютерных программах или пользователями баз данных для получения, вставки, удаления или изменения данных в базах данных:

- ● ЯМД
- ○ ЯОД
- ○ SQL
- ○ СУБД

55. Какая команда предназначена для изменения данных в таблицах?

- ○ Change
- ○ Direct
- ● Alter
- ○ Turn

56. В каком году был принят первый стандарт SQL?

- ○ 1992
- ○ 1978
- ○ 1999
- ● 1986

57. Какая команда предназначена для удаления таблицы?

- ○ DELETE TABLE
- ● DROP TABLE
- ○ KILL TABLE
- ○ CRASH TABLE

58. Можно ли создать таблицу, используя бланк QBE?

- ○ Да
- ● Нет
- ○ Зависит от версии MS Access

59. ::Тип SQL:: Какой тип SQL применяется для выполнения действий непосредственно в БД, чтобы получить результат?

- ● Интерактивный

60. Язык SQL позволяет выполнить следующие виды запросов (отметьте три варианта):

- ● **33%** Обновления данных
- ● **33%** Создания таблицы
- ○ **-100%** Отправка данных
- ● **34%** Управление БД

61. Какие типы языка SQL из перечисленных существуют (отметьте два варианта)?

- -100% Внешний
- 50% Встроенный
- 50% Интерактивный
- -100% Интегрированный

62. Какая команда служит для уничтожения представлений?

- DROP VIEW
- DELETE VIEW
- DROP LOOK
- DELETE KEY

Работа с современными промышленными СУБД/Организация данных в СУБД

63. При физической организации данных записи в файлах могут быть (отметьте два варианта):

- 50% Фиксированной длины
- -100% Произвольной адресации
- 50% Произвольной длины
- -100% Последовательной адресации

64. Для файлов последовательного доступа применяются следующие механизмы позиционирования (отметьте два варианта):

- -100% Указание индексов
- 50% Указание в начале записи ее длины
- 50% Указание специальным образом конца записи
- -100% Все варианты неверны

65. Недостатки метода хэширования (отметьте два варианта):

- 50% Возможность ситуации, когда для разных ключей будет получено одно и то же значение хэш-функции
- -100% Возможность ситуации, когда для разных ключей будут получены разные значения хэш-функции
- 50% Не всегда удается найти подходящую хэш-функцию
- -100% Значение хэш-функции зависит от первичного ключа

66. Несмотря на высокую эффективность метода хэширования, не всегда удается найти подходящую хэш-функцию. Поэтому для организации доступа по первичному ключу часто используют:

- Индексные файлы
- Файлы прямого доступа
- Функцию, однозначно связывающую номер записи и значение первичного ключа

- Файлы последовательного доступа

67. Достоинства хранения информации СУБД на «чистых» дисках (отметьте два варианта):

- -100%Использование стандартных средств обслуживания файлов
- 50%Внешняя память используется более эффективно
- 50%Как правило, увеличивается производительность обмена с дисками
- -100%Все варианты неверны

68. Достоинства работы СУБД с дисками через файловую систему (отметьте два варианта):

- -100%Экономия внешней памяти
- 50%В некоторых случаях выполнение операций ввода/вывода через файловую систему обеспечивает оптимизацию, которую СУБД не может реализовать
- 50%Использование файловой системы обладает большей гибкостью
- -100%Все варианты верны

69. Независимость данных – это:

- Неискажение данных при работе в многопользовательском режиме и в распределенных базах данных
- Адекватность отображения данных соответствующей предметной области
- Возможность изменения логической и физической структуры БД без изменения представлений пользователей
- Устойчивость хранимых данных к разрушению и уничтожению

70. Независимость данных (отметьте два варианта):

- 50%Предполагает инвариантность к характеру хранения данных
- -100%Предполагает защиту от ошибок при обновлении БД
- -100%Обеспечивает совместное использование данных многими пользователями
- 50%Обеспечивает минимальные изменения структуры БД при изменениях стратегии доступа к данным

71. Нарушение целостности данных может быть вызвано:

- Совместным выполнением конфликтных запросов пользователей
- Ошибками санкционированных пользователей или умышленные действия несанкционированных пользователей
- Программными сбоями СУБД или ОС
- Все варианты верны

72. Защита данных от несанкционированного доступа может достигаться (отметьте два варианта):

- 50%Получением разрешений от администратора базы данных

- -100%Защитой от ошибок при обновлении БД
- 50%Формированием видов - таблиц, производных от исходных и предназначенных конкретным пользователям
- -100%Стандартизацией построения и эксплуатации БД

Работа с современными промышленными СУБД/Методы поиска и анализа информации в базе данных

73. Какое выражение позволяет выполнить сортировку в обратном порядке?

- Order by desc;
- Sort by desc;
- Order by asc;
- Sort by asc;

74. Чем характеризуется каждая запись в таблице?

- Типом данных
- Своим порядковым номером
- Номером строки

75. Куда добавляется новая запись в таблице?

- В начало
- В конец
- По выбору
- Не добавляется

76. Упорядоченный список содержимого столбцов или группы столбцов - это?

- Дерево
- Индекс
- Модуль
- Адрес

77. Недостатком индексирования является?

- Индексирование проходит очень медленно
- Таблицу индексов необходимо перестраивать всякий раз, когда происходит изменение данных в таблице
- Индексирование проходит по дереву индексов
- При создании составного индексного файла индексирование выполняется не по одному, а по нескольким полям

78. Укажите, признаки соответствующие работе индексирования (отметьте три варианта)

- 33%Индексирование применяется для ускорения поиска
- -100%Индексирование замедляет работу поиска

- **33%**Простой индексный файл содержит в себе выровненное двоичное дерево
- **34%**Применение индексирования позволяет вести ускоренный поиск по разным полям таблицы, если для них построены индексы
- **-100%**Таблицу индексов не нужно перестраивать всякий раз, когда происходит изменение данных в таблице

79. Как организовано хранение индексов в MS Access и в FoxPro?

- В FoxPro для каждой таблицы индексов создаётся отдельный файл, а в MS Access индексы хранятся в том же файле, что и таблицы данных
- В MS Access для каждой таблицы индексов создаётся отдельный файл, а в FoxPro индексы хранятся в том же файле, что и таблицы данных
- В MS Access и FoxPro индексы хранятся в том же файле, что и таблицы данных
- В MS Access и FoxPro для каждой таблицы индексов создаётся отдельный файл

80. Укажите существующие виды индексов?

- Простые и сложные
- Кластерные и некластерные
- Ограниченные и неограниченные
- Последовательные и непоследовательные

81. На что необходимо обратить внимание при создании индекса?

- На дополнительную затрату ресурсов компьютера на сопровождение индекса
- На производительность системы в целом
- На удобность и надежность эксплуатации
- На модель данных

82. Что такое редкий индекс?

- Это файл с указателем
- Это файл с последовательностью пар ключей
- Это файл с последовательностью пар ключей и указателей
- Это файл с ключом

83. По какой команде назначается привилегии на доступ к собственной таблице?

- WITH GRANT
- GRANT ON OPTION
- SELECT FROM
- GRANT SELECT TO

84. Какая из привилегий даёт право модифицировать структуру таблицы?

- ALTER
- INSERT
- GRANT

- REFERENTS

85. Что позволяет выполнять привилегия EXECUTE?

- Предоставляет право создавать таблицы и индексы
- Пользователь с этой привилегией может удалить строки из таблицы
- Позволяет выполнять хранимую процедуру или функцию
- Пользователь с этой привилегией может выполнять запросы для таблицы
- Разрешает пользователю выполнять действия администратора базы данных, т.е. распоряжаться ею как своей собственной.

86. Какая команда осуществляет выборку информации?

- SELECT <список полей>
- <имя поля> IN <назначение>
- <имя поля> LIKE «шаблон»
- FROM <список таблиц>

87. Какой параметр определяет условное выражение, которое должно удовлетворять строки включаемые в таблицу результат?

- BETWEEN
- LIKE
- DISTINCT
- WHERE

Работа с современными промышленными СУБД/СУБД MS Access. Работа в интерактивном режиме

88. Выберите пункт, в котором соотношение тип поля – размер задано неверно:

- Текстовый – 255 байт
- Дата/время – 8 байт
- Денежный – 4 байта
- Memo – 64 000 байт

89. Какой тип поля обеспечивает связь и внедрение объектов, созданных в других приложениях?

- Счётчик
- Memo
- Объект
- Текстовый

90. Что такое маска поля?

- Описательное имя, которое будет использоваться вместо имени поля в отчётах
- Значение, которое присваивается полю при создании новой записи

- Шаблон, определяющий формат ввода значения поля
- Логическое выражение которое проверяется при вводе и редактировании поля

91. От чего зависят дополнительные параметры и ограничения, устанавливаемые для полей?

- От типа поля
- От связи между таблицами
- От заданных пользователем настроек
- От версии СУБД

92. Какой из типов связей является основным?

- 1:1
- 1:∞
- ∞:∞
- Все вышеперечисленные типы являются актуальными

93. Какой вид связи не допускается в реляционных БД?

- 1:1
- 1: ∞
- ∞:∞
- Все перечисленные виды допускаются

94. Что в MS Access является объектом для хранения данных?

- Запись
- Строка
- Таблица
- Столбец

95. Что является обязательным условием при создании связей между полями?

- Типы связываемых полей должны быть одинаковыми
- Связываемые поля должны иметь тип счётчик
- Связываемые поля должны иметь одинаковые имена
- Связываемые поля должны находиться в одной таблице

Работа с современными промышленными СУБД/СУБД MS Access. Связывание и фильтрация данных

96. С помощью какого инструмента наиболее удобно заполнять базу данных в MS Access?

- Отчёт
- Форма
- Конструктор

- Мастер

97. Каким образом работает Фильтр по выделенному?

- Сохраняет в файл выделенные записи
- На экран выводится пустая таблица или форма для активного объекта базы данных
- Извлекает из таблицы и выдает на экран только те записи, которые содержат выделенное значение
- В верхней части выводится список полей активной таблицы. В нижней части окна выводится бланк запроса. В строку бланка запроса Поле перетаскиваются мышью из списка поля, по которым надо задать условия отбора записей. Условия отбора вводятся в соответствующую строку. Кроме того, в бланке запроса в строке Сортировка может быть выбран тип сортировки для одного или нескольких выбранных полей

98. Каким образом работает Обычный фильтр?

- После выполнения команды Записи|Фильтр|Изменить фильтр в окне обычного фильтра Фильтр на экран выводится пустая таблица или форма для активного объекта базы данных. На вкладке Найти в поля фильтра вводятся значения, по которым будут отбираться записи. Ввод значений в несколько полей одной строки фильтра определяет отбор записей, в которых присутствуют все указанные значения. При этом заданные условия рассматриваются как объединяемые логической операцией "И"
- Извлекает из таблицы и выдает на экран только те записи, которые содержат выделенное значение
- Обычно работает
- Такого фильтра нет

99. Какие три вида фильтров предусмотрены в MS Access?

- **33%** Фильтр по выделенному
- **33%** Обычный фильтр
- **-100%** Масляный фильтр
- **34%** Расширенный фильтр
- **-100%** Фильтр с параметрами
- **-100%** Фильтр Гейтса

100. Каким образом можно получить доступ к инструменту Схема данных?

- Вид|Схема данных
- В окне Базы данных с выполнения команды Сервис|Схема данных
- Файл|Схема данных
- Такого инструмента нет в среде MS access

101. Сколько листов данных, за одну операцию импорта, можно импортировать из Exel в Access?

- Один
- Два

- Три
- Четыре

102. Какие виды связей между таблицами, можно создавать в MS Access? (отметьте два варианта)

- **50%** 1 к 1
- **50%** 1 к ∞
- **-100%** 1 к 2
- **-100%** 1 к 50

103. Как можно распечатать данные из базы данных?

- С использованием отчёта
- Через конструктор
- В MS Access нельзя вывести на печать документ
- Необходимо экспортировать данные в MS Word и только потом распечатать

104. Продолжите предложение. Если поле связи является уникальным ключом в одной таблице (главной таблицы связи), а в другой таблице (подчиненной таблице связи) является не ключевым или входит в составной ключ, то его значения …..

- Не могут повторятся
- Равны нулю
- Могут повторяться
- Суммируются

105. Технология связывания и внедрения объектов в другие документы и объекты, разработанные компанией Microsoft?

- PDM
- OLE
- OSPF
- STP

Работа с современными промышленными СУБД/Средства диалогового постороения запросов

106. Назовите основные преимущества перекрестных запросов:

- Возможность сортировки таблицы результатов по значениям, содержащимся в столбцах; простота и скорость разработки сложных запросов с несколькими уровнями детализации
- Простота и скорость разработки сложных запросов с несколькими уровнями детализации; возможность обработки значительного объема данных и вывода их в формате, который очень хорошо подходит для автоматического создания графиков и диаграмм
- Возможность обработки значительного объема данных и вывода их в формате, который очень хорошо подходит для автоматического создания графиков и диаграмм; простота построения отчётов

- Возможность сортировки таблицы результатов по значениям, содержащимся в столбцах; простота построения отчётов

107. Что такое схема данных?

- Схема: наглядно отображающая таблицы и связи между ними; обеспечивающая использование связей при обработке данных
- Схема, наглядно отображающая запросы и отчёты
- Схема: обеспечивающая использование связей при обработке данных; наглядно отображающая запросы
- Схема: наглядно отображающая таблицы и связи между ними; отображающая отчёты

108. Что позволяют выполнять запросы действий?

- Создавать таблицы, модифицировать данные в таблицах: удалять, обновлять, добавлять записи
- Удалять и добавлять данные в таблицах
- Создавать таблицы без возможности модифицировать данные в них
- Обновлять и добавлять записи в таблицах

109. Какие запросы относятся к запросам группы действий?

- Запросы на: создание таблиц, добавление, обновление, удаление записей в таблицах
- Запросы на создание отчётов
- Запросы на создание таблиц и отчётов
- Запросы на создание отчётов и удаление записей в таблицах

110. Что нужно сделать для того, чтобы создать многотабличный запрос?

- В окно конструктора запросов добавить все участвующие в выборке таблицы, определить условия объединения таблиц
- В окно конструктора запросов можно не добавлять все участвующие в выборке таблицы, но обязательно определить условия объединения таблиц
- В окно конструктора запросов добавить все участвующие в выборке таблицы
- В окно конструктора запросов добавить любой существующий запрос на выборку

111. Какая конструкция используется для группировки данных в запросе?

- Group by
- Order by
- Where
- Count
- Select

112. Какая строка отсутствует в QBE бланке:

- Поле
- Условия отбора
- Сортировка

- Обновление данных

113. Выберите верное высказывание:

- QBE - запрос по образцу – средство для отыскания необходимой информации в базе данных
- QBE – запрос формируется на специальном языке
- Все запросы Access строит на основе QBE – запросов
- В MS Access применяется только один тип запросов: по образцу (QBE – Query by example)

114. Какие поля необходимо определить для перекрестного запроса?

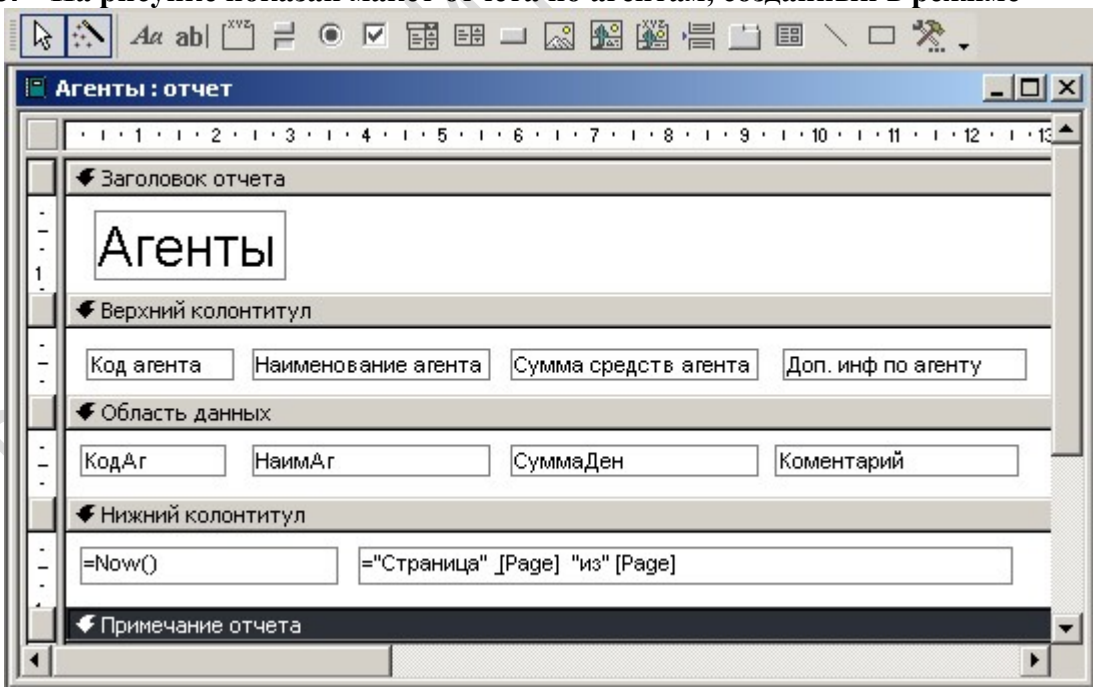
- Одно поле в качестве заголовков строк, одно для заголовков столбцов и одно поле значений
- Два поля в качестве заголовков строк, два для заголовков столбцов и одно поле значений
- Одно поле в качестве заголовков строк, одно для заголовков столбцов
- Одно поле в качестве заголовков строк, два для заголовков столбцов и одно поле значений

115. Чтобы открыть схему данных, необходимо выполнить команду:

- Сервис, Схема данных
- Вставка, Схема данных
- Вставка, Таблица, Схема данных
- Правка, Вставить, Схема данных

Работа с современными промышленными СУБД/СУБД MS Access. Организация пользовательского интерфейса

116. На рисунке показан макет отчета по агентам, созданный в режиме



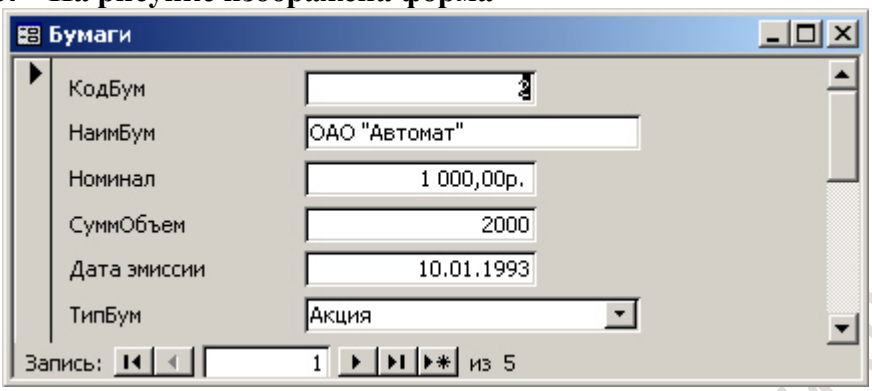
- Автоотчет ленточный
- Автоотчет в столбец
- Автоотчёт табличный

- Нет правильного ответа

117. Кнопка "Конструктор" открывает

- Структуру объекта
- Содержимое таблицы
- Панель элементов
- Выводит на печать таблицу

118. На рисунке изображена форма



- В столбец
- Ленточная
- Табличная
- Нет правильного ответа

119. Какого раздела не существует в конструкторе форм?

- Заголовка
- Верхнего колонтитула
- Примечание
- Итоговый

120. При помощи какой панели устанавливаются кнопки в форме?

- Стандартная
- Кнопки
- Элементов
- Форматирования

121. Из чего состоит макрос?

- Из набора тегов
- Из совокупности операторов Visual Basic
- Из набора гиперссылок
- Из набора макрокоманд

122. Кнопочная форма Access создаётся:

- Конструктором форм

- Мастером форм
- Автоформами
- Редактором форм

123. Переключатели в форме устанавливаются с помощью панели:

- Стандартная
- Формы
- Элементов
- Переключателей

124. Мастер отчётов -

- Запускает основного мастера для создания отчетов, позволяющего выбрать поля для отчета, задать форматы, условия группировки и итоговые функции
- Помогает построить диаграмму и создает в отчете свободную рамку объекта OLE с внедренной диаграммой Microsoft Graph
- Позволяет создавать отчет «с нуля» и редактировать уже созданный отчет
- Нет правильного ответа

125. В Microsoft Access 2000 содержится список макрокоманд, сгруппированных по категориям:

- Работа с данными в формах и отчетах
- Выполнение команд, макросов, процедур и запросов
- Работа с объектами
- Всё перечисленное

Функциональные зависимости и поиск данных/Функциональные зависимости на данные

126. В процессе чего следует избавиться от всех "других" функциональных зависимостей, т.е. от тех, которые имеют иной вид, чем K=>F (K - первичный ключ, а F - некоторое другое поле)?

- Составления схемы данных
- Нормализации
- Создания отчета
- Нет правильного ответа

127. Что такое представление?

- Файл с последовательностью пар ключей и указателей на запись в файле данных
- Виртуальная (логическая) таблица, представляющая собой поименованный запрос
- Объект базы данных, создаваемый с целью повышения производительности поиска данных
- Нет правильного ответа

128. Что такое индекс?

- Объект базы данных, создаваемый с целью повышения производительности поиска данных
- Подмножество атрибутов отношения, удовлетворяющее требованиям уникальности и минимальности (несократимости)
- Оба верны
- Нет правильного ответа

129. Что такое редкий индекс?

- Файл с последовательностью пар ключей и указателей
- Файл с последовательностью пар ключей и указателей на запись в файле данных
- Оба верны
- Нет правильного ответа

130. Какой первичный ключ называется простым?

- Первичный ключ состоит из трех атрибутов
- Первичный ключ состоит из двух атрибутов
- Первичный ключ состоит из единственного атрибута
- Нет правильного ответа

131. Какой первичный ключ называется составным?

- Первичный ключ состоит из трех атрибутов
- Первичный ключ состоит из двух и более атрибутов
- Первичный ключ состоит из единственного атрибута
- Нет правильного ответа

132. Выберите из предложенных вариантов аксиому аддитивности:

- Если X =>Y и Y=>Z, то X=>Z (можно выявлять неявные зависимости)
- Если X=>YZ, то X=>Y (можно отрезать атрибуты справа)
- Если X=>Y и X=>Z, то X=>YZ (можно объединять зависимости с одинаковыми левыми частями)
- Нет правильного ответа

133. Выберите из предложенных вариантов аксиому транзитивности:

- Если X =>Y и Y=>Z, то X=>Z (можно выявлять неявные зависимости)
- Если X=>YZ, то X=>Y (можно отрезать атрибуты справа)
- Если X=>Y и X=>Z, то X=>YZ (можно объединять зависимости с одинаковыми левыми частями)
- Нет правильного ответа

134. Какой граф считается ациклическим?

- Граф, содержащий один цикл
- Граф, не содержащий ни одного цикла
- Граф, содержащий два и более циклов

- Нет правильного ответа

135. Какие зависимости бывают?

- Однозначные
- Многозначные
- Оба верны
- Нет правильного ответа

Функциональные зависимости и поиск данных/Проблема аномалии и задача нормализации данных

136. Причинами аномалий при работе с данными являются:

- Хранение в одном отношении разнородной информации
- Схема отношения выполнена с соблюдением правил нормализации
- Избыточность данных, также порожденная тем, что в одном отношении хранится разнородная информация
- Каждый неключевой атрибут зависит от ключевого

137. Какие действия должны быть произведены для устранения всех типов аномалий (ввода,удаления,обновления)?

- Нормализация исходных схем
- Индексация схем
- Организация ссылочной целостности
- Закрытие прав доступа на работу с таблицей посторонним лицам

138. Нормальная форма — свойство отношения в реляционной модели данных, (…), которая потенциально может привести к логически ошибочным результатам выборки или изменения данных.

- Характеризующее его с точки зрения избыточности
- Характеризующее его с точки зрения надежности
- Характеризующее его с точки зрения упрощённости
- Характеризующее его с точки зрения аутентичности

139. Отношение находится в BCNF(нормальная форма Бойса-Кодда) тогда и только тогда, когда каждая ее нетривиальная и неприводимая слева функциональная зависимость имеет в качестве своего детерминанта:

- Сложный кортеж
- Потенциальный индекс
- Некоторый потенциальный ключ
- Строку состояния

140. Нормальная форма Бойса-Кодда представляет собой более строгую версию:

- Второй НФ

- Третьей НФ
- Первой НФ
- Четвертой НФ

141. Отношение находится в 3 НФ тогда и только тогда, когда выполняются следующие условия (отметьте два варианта):

- **50%** Отношение находится во второй нормальной форме
- **-100%** Отношение находится в первой нормальной форме (1НФ)
- **-100%** Таблица не содержит нетривиальных многозначных зависимостей
- **50%** Каждый неключевой атрибут отношения находится в нетранзитивной (то есть прямой) зависимости от потенциального ключа

142. Отношение находится в первой нормальной форме (1НФ) тогда и только тогда:

- Когда таблица не содержит нетривиальных многозначных зависимостей
- Когда в любом допустимом значении отношения каждый его кортеж содержит только одно значение для каждого из атрибутов
- Когда в любом допустимом значении отношения каждый его кортеж может содержать несколько значений для каждого из атрибутов

143. Общее назначение процесса нормализации заключается в следующем:

- Исключение некоторых типов избыточности
- Устранение некоторых аномалий обновления
- Упрощение процедуры применения необходимых ограничений целостности
- Разработка проекта базы данных, интуитивно понятного и достаточно качественного, для возможности его дальнейшего расширения
- Верны все варианты ответов

144. Конечной целью нормализации является:

- Уменьшение физического объёма БД
- Увеличение физического объёма БД
- Уменьшение потенциальной противоречивости хранимой в БД информации
- Увеличение производительности работы

145. Какой процесс называется нормализацией?

- Процесс устранения внутренних аномалий БД
- Процесс преобразования отношений базы данных к виду, отвечающему нормальным формам
- Процесс преобразования базы данных к виду, наиболее близкому к НФ Бойса-Кодда
- Любой процесс увеличивающий избыточность данных

Функциональные зависимости и поиск данных/Использование функциональных зависимостей при работе с данными

146. В каких случаях применяется функция хэширования (отметьте три варианта):

- -100% Удаление данных
- 33% Проверка на наличие ошибок
- -100% Объединение данных в массивы и последующая сортировка
- 33% Сверка данных
- 34% Ускорение поиска данных

147. Какая сортировка чаще всего используется в СУБД:

- Топологическая сортировка
- Блинная сортировка
- Внешняя сортировка
- Сортировка Гаппо

148. Основная характеристика вычислительной сложности алгоритма:

- Время
- Объем алгоритма
- Наличие вложенных циклов
- Количество требуемой памяти

149. Укажите утверждение не характерное для ассоциативного поиска в массиве:

- Для ускорения операции поиска можно упорядочить элементы массива (ассоциативный массив) по ключу и осуществлять поиск методом деления пополам
- Для поиска в ассоциативном массиве невозможно использовать хэш-таблицы
- Использование деревьев поиска в ассоциативном массиве

150. Так как физические записи имеют разную длину, то при модификации данных запись может увеличиться и превысит исходную длину записи до модификации. В этом случае при определенных методах хранения может понадобиться дополнительное пространство хранения, где и будут размещены дополнительные данные. Это пространство называется:

- Добавочное пространство
- Область переполнения
- Внешняя область
- Внутренняя область

151. ::Оператор:: Какой оператор в SQL определяет, совпадает ли указанная символьная строка с заданным шаблоном?

- like

152. Какой комбинацией клавиш в MS Access вызывается окно поиска?

- CTRL+F
- CTRL+U

- CTRL+ALT+W
- CTRL+H

153. ::Ключевое слово:: Напишите, какое слово необходимо использовать для сортировки по убыванию:
SELECT DISTINCT Predmety.Predm
FROM Predmety
ORDER BY Predmety.Predm [СЛОВО];

- desc

154. Может ли сортировка в MS Access выполняться по нескольким столбцам?

- Затрудняюсь ответить
- Нет
- Да

Функциональные зависимости и поиск данных/Языки баз данных

155. Выберите правильное понятие языка манипулирования данными:

- Первый язык программирования высокого уровня, имеющий транслятор
- Язык разметки гипертекста
- Семейство компьютерных языков, используемых в компьютерных программах или пользователями баз данных для получения, вставки, удаления или изменения данных в базах данных
- Язык и система программирования, предназначенная для поддержки начальных курсов информатики и программирования в средней и высшей школе

156. Укажите наиболее популярный на текущий момент язык DML:

- ЛИНТЕР
- SQL
- MongoDB
- CouchDB

157. Укажите преимущество SQL:

- Лёгкая возможность перенести тексты SQL-запросов из одной СУБД в другую
- Отсутствие поддержки свойства «=»
- Неопределённые значения (nulls)
- Повторяющиеся строки

158. Укажите год первого варианта стандарта SQL, принятый институтом ANSI:

- 1986
- 1989
- 1992
- 1999

159. Выберите объявления, относящиеся к точным числам (отметьте два варианта) :

- -100%VARCHAR
- 50%DECIMAL
- 50%SMALLINT
- -100%INTERVAL

160. Что определяет тип BLOB?

- Набор допустимых значений для одного или нескольких атрибутов
- Может хранить не более 65 535 символов
- Дробное число, хранящееся в виде строки
- Число с плавающей точкой двойной точности

161. Какой запрос используется для получения информации, хранящейся в базе данных?

- DELETE
- UPDATE
- INSERT
- SELECT

162. Какой запрос используется для создания новой строки данных?

- UPDATE
- INSERT
- DELETE
- SELECT

163. Используя какой запрос можно удалить таблицы (TABLE), индексы (INDEX) и базы данных (DATABASE)?

- DROP
- UPDATE
- REVOKE
- ALTER TABLE

164. При формировании запроса SELECT пользователь описывает ожидаемый набор данных:

- Тип и значения данных
- Вид и содержимое
- Вид
- Содержимое

Функциональные зависимости и поиск данных/Использование языка запросов SQL для выборки данных

165. Дайте определение оператору SELECT

- Оператор DML языка SQL, возвращающий набор данных (выборку) из базы данных, удовлетворяющих заданному условию
- Оператор в SQL, указывающий, что оператор языка управления данными (DML) должен действовать только на записи, удовлетворяющие определенным критериям
- Оба варианта возможны
- Нет верного определения

166. Какие ключевые слова относятся к запросу SELECT?

- WHERE , HAVING
- GROUP BY
- ORDER BY
- Все варианты верны

167. SELECT — оператор, возвращающий набор данных из базы данных, удовлетворяющих заданному условию. В большинстве случаев, выборка осуществляется:

- Из одной таблицы
- Из нескольких таблиц
- Из одной или нескольких таблиц
- Все варианты неверны

168. Формат запроса с использованием оператора SELECT

- SELECT список полей FROM список таблиц WHERE условия…
- SELECT список полей GROUP BY список таблиц WHERE условия…
- SELECT список полей FROM список таблиц ORDER BY условия…
- Нет верного ответа

169. Следующее утверждение верно:

- Алиас - псевдоним для конкретной таблицы в некоторой БД
- Алиас - имя для абстрактной таблицы в некоторой БД
- Алиас - имя для таблицы в некоторой БД
- Все варианты неверны

170. При формировании запроса SELECT пользователь описывает ожидаемый набор данных:

- Тип и значения данных
- Вид и содержимое
- Вид
- Содержимое

171. Управление полями — это

- Указание полей таблицы (таблиц), исключаемые из результирующего набора данных
- Указание полей таблицы (таблиц), включаемых в компенсирующий набор данных
- Указание полей таблицы (таблиц), включаемых в результирующий набор данных
- Нет верных ответов

172. SELECT используется:

- Для отбора записей, удовлетворяющих сложным критериям поиска
- Для вывода, удовлетворяющих сложным критериям поиска
- Все варианты верны
- Все варианты неверны

173. Результатом выполнения оператора SELECT является:

- Действие
- Таблица
- Действие и таблица
- Все ответы неверны

174. Как выглядят с точки зрения оператора SELECT постоянно хранимые таблицы и временные таблицы?

- Одинаково
- Различно
- Различия скрыты от пользователя, но при реальном выполнении учитываются
- Все ответы неверны

Функциональные зависимости и поиск данных/Отбор и сортировка записей в запросах для выборки данных

175. При помощи какого ключевого слова в операторе Select можно исключить повторяющиеся записи?

- Join
- Having
- Distinct
- Like

176. Каким образом происходит выборка значений из нескольких таблиц?

- Нужно провести суммирование таблиц и далее провести запрос по получившейся таблице
- Достаточно указать поля, по которым идет отбор, а таблицу запрос определит сам
- Нужно указать таблицы в списке таблиц
- Нужно перед выборкой провести операцию реляционной алгебры проекция и по получившемуся множеству записей провести запрос

177. Какие операции можно применять в предложении Where?

- Любые
- Логические
- Математические
- Операции реляционной алгебры

178. Определение предложения Group by:

- Данное предложение позволяет определить множество значений отдельного поля в терминах другого поля и применить к ним функции агрегирования
- Данное предложение позволяет генерировать значения истина, если хоть одно из группируемых значений выдает истину
- Данное предложение позволяет генерировать значения ложь, если хоть одно из группируемых значений выдает ложь
- Данное предложение позволяет определить набор записей, к которым будет применятся функция агрегирования

179. Каково назначение оператора Like?

- Создает набор из допустимых числовых значений
- Является аналогом логической функции OR
- Является аналогом логической функции AND
- Задает текстовую строку с возможным использованием метасимволов

180. Каково назначение оператора Is null?

- Обнуляет таблицу, задавая всем ячейкам значения null
- Служит для сравнения с ячейками, значения которых null
- Служит для поиска значений null в таблице
- При указании этого оператора все столбцы и строки содержащие значения null удаляются

181. Аналогом какого логического выражения является оператор In?

- имя_поля = значение 1 or имя_поля = значение 2
- имя_поля >= значение 1 and имя_поля <= значение 2
- имя_поля like «шаблон»
- имя_поля not логического значение

182. Выберите правильное выражение (отметьте два варианта):

- **50%** Asc – сортировка по возрастанию
- **50%** Desc – сортировка по убыванию
- **-100%** Asc – сортировка по убыванию
- **-100%** Any – сортировка по убыванию

183. Аналогом какого логического выражения является оператор Between?

- имя_поля = значение 1 or имя_поля = значение 2
- имя_поля >= значение 1 and имя_поля <= значение 2

- имя_поля like «шаблон»
- имя_поля not логического значение

184. Для какой цели применяются функции агрегирования?

- Для вычисления операции разность реляционной алгебры
- Для составления подзапросов
- Для расчета значений по группам строк
- Для работы с индексами таблиц

Функциональные зависимости и поиск данных/Группировка данных в запросе

185. Процедура объединения в логическом порядке строк с определенными значениями это:

- Сортировка данных
- Группировка данных
- Суммирование данных
- Обобщение данных

186. Одним из видов группировки является сортировка (определяемая фразой …), кроме этого в группе условий оператора выборки SELECT могут быть использованы фразы группировки … и … . Выберите вариант ответа, в котором записаны фразы в порядке их замены троеточиями.

- ORDER BY, GROUP BY, HAVING
- HAVING, WHERE, COUNT
- ORDER BY, AVG, HAVING
- GROUP BY, ORDER BY, SUM

187. Фраза GROUP BY с синтаксисом: … позвляет преобразовать таблицу так, что в ее отдельные строки будет собрано содержание всех строк с одинаковыми значениями полей группировки. Выберите вариант ответа с правильным синтаксисом.

- GROUP BY SELECT имя_поля [,имя_поля ...]
- GROUP BY FROM имя_поля [,имя_поля ...]
- GROUP BY имя_таблицы [,имя_таблицы ...]
- GROUP BY имя_поля [,имя_поля ...]

188. Для каких полей используется группировка ?

- Используется для тех полей, по значениям которых нужно сгруппировать записи таблицы
- Используется для тех полей, значения которых не надо группировать
- Используется для тех полей, по значениям которых нужно удалить записи таблицы
- Используется для пустых полей

189. Функции COUNT, MAX, MIN, SUM, AVG предназначенные для расчета значений по группам строк таблиц называются:

- Функции группировки
- Функции множеств
- Функции агрегирования
- Функции объединения

190. Выберите варианты ответа с правильными запросом (отметьте два ответа).

- **50%**SELECT COUNT(stock) FROM stock WHERE stock <> 0
- **-100%**SELECT * FROM tab1 WHERE col1 = MAX(col1)
- **50%**SELECT AVG(OCEN) FROM ОЦЕНКА WHERE SNUM=”003”
- **-100%**SELECT SUM(FAMILIA) FROM FIO WHERE SNUM=АБВ

191. Из функций агрегирования только с числовыми полями могут работать?

- COUNT, MAX
- MAX, MIN, SUM, AVG
- SUM, AVG
- SUM

192. Из функций агрегирования работать с числовыми и символьными полями могут?

- MAX, MIN, SUM, AVG
- MAX, SUM, AVG
- MAX, MIN
- COUNT, MAX, MIN

193. Выберите варианты ответов с корректными запросами (отметьте два ответа):

- **50%**SELECT OSNUM, AVG(OCEN) FROM OCENKA HAVING AVG(OCEN)>=4
- **-100%**SELECT OSNUM, AVG(OCEN) FROM OCENKA WHERE AVG(OCEN)>=4 GROUP BY OSNUM
- **-100%**SELECT OSNUM, MIN(OCEN) FROM OCENKA GROUP BY OSNUM HAVING ODATE > 14/01/10
- **50%**SELECT OSNUM, MIN(OCEN) FROM OCENKA GROUP BY OSNUM WHERE ODATE > 14/01/10

194. Какой аргумент группировки должен иметь единственное значение для каждой выходной группы?

- COUNT
- HAVING
- SELECT
- GROUP BY

Функциональные зависимости и поиск данных/Выборка данных из множества таблиц

195. Выберите верное утверждение:

- Самообъединение – объединение таблицы с этой же таблицей – является другим вариантом объединения таблиц на основе операции эквисоединения. В этом случае сравниваются значения внутри столбца одной таблицы
- Самообъединение объединяет вывод двух или более SQL запросов в единый набор строк и столбцов
- Самообъединение позволяет определять множество значений отдельного поля в терминах другого поля
- Самообъединение позволяет вкладывать запросы друг в друга

196. Укажите пример без декартового произведения:

- SELECT Меню.*, Трапезы.*, Вид_блюд.*, Блюда.*
  FROM Меню, Трапезы, Вид_блюд, Блюда;
- SELECT Меню.*, Трапезы.*, Вид_блюд.*, Блюда.*
  FROM Меню, Трапезы, Вид_блюд, Блюда
  WHERE Меню.Т = Трапезы.Т
  AND Меню.В = Вид_блюд.В
  AND Меню.БЛ = Блюда.БЛ;
- SELECT Вид_блюд.*, Трапезы.*
  FROM Вид_блюд, Трапезы;
- SELECT блюда
  FROM Вид_блюд WHERE Вид_блюд = салаты

197. Операция Left Join:

- Объединяет записи двух таблиц при выполнении условий
- Создает левое внешнее объединение, при котором все записи из первой таблицы включаются в результат, даже если со второй правой таблицы нет соответствия по условию записей
- Создает правое внешнее объединение, при котором все записи из второй таблицы включаются в результат, даже если с первой левой таблицы нет соответствия по условию записей
- Нет правильного варианта

198. Самым распространенным типом объединения можно считать:

- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- Все они одинаково распространены

199. Выберите верные утверждения (отметьте два варианта):

- **-100%**Операция INNER JOIN может включаться в операцию LEFT JOIN
- **50%**Операция LEFT JOIN может включаться в операцию INNER JOIN
- **50%**Операция RIGHT JOIN может включаться в операцию INNER JOIN
- **-100%**Операция INNER JOIN может включаться в операцию RIGHT JOIN

200. Выберите связующее поле:
SELECT ИмяКатегории, Наименование, Время, Сложность
FROM Категории INNER JOIN Товары
ON Категории.ИДКатегории = Товары.ИДКатегории;

- ИДКатегории
- Наименование
- Время
- Сложность

201. Выберите два возможных оператора:
…FROM таблица1 INNER JOIN таблица2 ON таблица1.поле1 ВОЗМОЖНЫЙ ОПЕРАТОР таблица2.поле2…

- **50%** =
- **-100%** AND
- **-100%** OR
- **50%** <>

202. SELECT Вид_блюд.*, Трапезы.*
FROM Вид_блюд, Трапезы;
В результате этого запроса, если видов блюд 5, а трапез 3, мы получим таблицу, содержащую:

- 8 строк
- 15 строк
- 2 строки
- 5 строк

***203.* SELECT sname, pname**
FROM преподаватель, студент
WHERE pnum=spdp
GROUP BY sname DESC;
Это SQL запрос на выборку данных из множества таблиц. Результатом этого запроса будут:

- Все строки, входящие в таблицы: преподаватель, студент
- Строки, входящие в таблицу преподаватель и не входящие в таблицу студент
- Строки таблиц преподаватель и студент, в которых поля pnum и spdp не совпадают
- Строки таблиц преподаватель и студент, в которых поля pnum и spdp совпадают

204. Менее ресурсоемким типом объединения можно считать:

- CROSS JOIN
- LEFT JOIN
- FULL JOIN
- Все варианты одинаково ресурсоемки

Функциональные зависимости и поиск данных/Использование подзапросов в языке запросов SQL

205. Какой из команд использующей подзапросы можно изменить содержание строк в базе данных?

- ○ DELETE
- ● UPDATE
- ○ INSERT
- ○ SELECT

206. Как включаются подзапросы в приложении WHERE? (отметьте три варианта)

- ● 33%С помощью условия IN
- ● 33%С помощью условия EXISTS
- ○ -100%Записываются через запятую
- ● 34%С помощью условий сравнения (= | <> | < | <= | > | >=)

207. Определите порядок проведения процедура оценки, выполняемой связанным запросом:
1.Выполнить подзапрос. Для отбора записей использовать строку-кандидат.
2.Выбрать строку из таблицы, указанной во внешнем запросе. Это будет текущая строка-кандидат.
3.Вычислить условие во внешнем запросе, на основе результатов внутреннего подзапроса, выполняемого в п.3. Определяется - отбирается ли строка-кандидат для вывода.
4.Повторить процедуру для всех строк.
5.Сохранить значения из этой строки-кандидата во временном буфере.

- ○ 1, 2, 3, 4, 5
- ○ 2, 4, 1, 3, 5
- ● 2, 5, 1, 3, 4
- ○ 3, 2, 1, 4, 5

208. Из приведенный ниже подзапросов выберите связанный запрос.

- ○ SELECT COUNT(*) AS orders
  FROM Orders
  WHERE cust_id = '1000000001';
- ● SELECT *
  FROM Customers C
  WHERE '1999-10-03' IN (
  SELECT odate
  FROM Orders O
  WHERE O.cnum = C.cnum)
- ○ SELECT cust_name, cust_state,(SELECT COUNT(*)
  FROM Orders
  WHERE Orders.cust_id = Customers.cust_id) AS orders FROM Customers ORDER BY cust_name;

- SELECT *
  FROM Orders outer
  WHERE amt >
  (SELECT AVG amt
  FROM Orders inter
  WHERE inner.cnum = outer.cnum);

209. Как происходит связывание таблицы со своей копией?

- При помощи выражения WHERE
- При помощи выражения HAVING
- Командой SELECT
- Командой FROM

210. Какой из операторов может вывести NULL-значения?

- ANY
- ALL
- EXISTS
- SOME
- NOT EXISTS

211. Оператор EXISTS используется:

- Чтобы подзапрос не выводился
- Чтобы произвести вывод подзапроса
- Чтобы указать предикату, производить ли подзапросу вывод или нет
- Для всего перечисленного

212. Чем отличается ANY от EXISTS? (отметьте два варианта)

- **-100%**Подзапрос не выбирает значения такого же типа как и те, которые сравниваются в основном предикате
- **50%**Подзапрос должен выбирать значения такого же типа как и те, которые сравниваются в основном предикате
- **50%**Нет возможности работать со значениями NULL
- **-100%**Есть возможность работать со значениями NULL

213. Операторы SOME и ANY:

- Взаимозаменяемы везде
- Не взаимозаменяемы
- Использоваться только с подзапросами
- Все варианты не верны

214. С помощью ALL предикат будет верным?

- Если каждое значение, выбранное подзапросом, удовлетворяет условию в предикате внешнего запроса

- Если каждое значение, выбранное подзапросом, не удовлетворяет условию в предикате внешнего запроса
- Если каждое значение, выбранное подзапросом, удовлетворяет условию в предикате внутреннего запроса
- Если каждое значение, выбранное подзапросом, не удовлетворяет условию в предикате внутреннего запроса

++

Функциональные зависимости и поиск данных/Объединение запросов

215. Какое из перечисленных ниже предложений является объединением?

- HEAVING
- UNION
- GROUP BY
- WHERE
- UPDATE

216. Какую роль выполняет предложение UNION?

- Определяет критерий включения записей по группам в результат
- Сортирует записи, возвращенные запросом, по возрастанию или по убыванию значений указанного поля (полей)
- Группирует указанному перечню столбцов с тем, чтобы получить для каждой группы единственное агрегированное значение
- Выбирает данные из указанных столбцов и (если необходимо) выполнить перед выводом их преобразование в соответствии с указанными выражениями и (или) функциями
- Позволяет получить отношение, состоящее из всех строк, входящих в одно или оба объединяемых отношения

217. Какие существуют условия совместимости запросов для объединения? (отметьте два варианта)?

- **50%** Каждый запрос должен указывать одинаковое число столбцов
- **-100%** Каждый запрос должен указывать одинаковое число строк
- **50%** Когда пустые значения(NULL) запрещены в любом столбце объединения
- **-100%** Запросы должны иметь одинаковое число подзапросов

218. Какое количество запросов позволяет объединять UNION?

- 2 запроса
- Не более 3-х
- До 13 запросов
- Любое число запросов

219. В чём отличие предложения Union от подзапросов?

- В нем ни один из двух (или больше) запросов не управляются другим запросом
- В нем запросы управляются одним из запросов запросов
- Ничем

220. Как можно задать объединение таблиц?

- При помощи предложения UNION
- При помощи JOIN
- При помощи ORDER BY

221. Что такое Union?

- Предложение
- Запрос
- Выражение
- Таблица
- Подзапрос

222. Получить повторяющиеся строки в МА можно с помощью:

- Union ALL
- NOT NULL
- Union NOT NULL

223. Выберете неправильное выражение:

- SELECT osnum, MIN(ocen)
  FROM оценка GROUP BY osnum
  UNION
  SELECT osnum, MAX(ocen)
  FROM оценка GROUP BY osnum;
- SELECT pnum, pname FROM преподаватель
  WHERE pnum IN
  (SELECT DISTINCT spdp FROM студент)
  UNION
  SELECT snum, sname FROM студент
  WHERE spdp IS NOT NULL;
- SELECT pnum
  UNION преподаватель;

224. Для описания множеств, получающихся при пересечении и объединении таблиц, какой используется специальный математический аппарат?

- Реляционная алгебра
- Аналитическая геометрия
- Математический анализ

Функциональные зависимости и поиск данных/Запросы обновления таблиц в языке запросов SQL

225. Выберите правильный формат команды обновления данных:

- ○ INSERT INTO список_значений [(имя_таблицы)] VALUES (список_имен)
- ● UPDATE имя_таблицы SET поле=значение [, поле=значение, …] WHERE условие
- ○ DELETE FROM имя_таблицы VALUES (список_значений)

226. Выберите утверждение, справедливое при работе с командой INSERT:

- ○ Согласно стандарту SQL, допускается, что вводимые значения могут быть выражениями
- ○ При указании списка имен полей их порядок должен совпадать с порядком полей в таблице
- ● Список значений должен указываться в порядке списка имен полей

227. Выберите неверный запрос:

- ○ INSERT INTO A SELECT a FROM B WHERE a=2
- ● INSERT INTO A SELECT * FROM A WHERE a=2
- ○ INSERT INTO A SELECT b FROM B WHERE a=2

228. Выберите правильный запрос на обновление поля a в таблице A на значение t при условии x:

- ○ UPDATE A SET x=t WHERE a
- ○ UPDATE a SET a=t WHERE x
- ○ UPDATE t SET A=a WHERE x
- ● UPDATE A SET a=t WHERE x

229. Выберите правильный запрос на обновление записей, где a, b – поля, A, B – таблицы, x - условие:

- ● UPDATE A SET a=1 WHERE a=(SELECT b FROM B WHERE x)
- ○ UPDATE A SET a=1 WHERE (SELECT b FROM B WHERE x)=a
- ○ UPDATE A SET a=1 WHERE a=(SELECT b FROM A WHERE x)

230. Выберите правильный запрос на удаление из таблицы A всех записей:

- ● DELETE FROM A
- ○ DELETE FROM A WHERE NULL
- ○ DELETE FROM A WHERE *

231. Как правильно удалять информацию из взаимосвязанных таблиц?

- ○ Удалить информацию из главной таблицы, затем из подчиненных
- ● Удалить информацию из подчиненных таблиц, затем из главной
- ○ Удалить информацию из главной таблицы

232. Где в команде DELETE могут использоваться подзапросы?

- ○ При выборе полей таблицы
- ○ В команде DELETE подзапросы не используются
- ● В условии отбора удаляемых записей

233. Где в команде UPDATE могут использоваться подзапросы:

- ○ При установке нового значения поля
- ● В условии отбора обновляемых записей
- ○ При выборе обновляемых таблиц

234. Что может использоваться в качестве значений полей при работе с командой UPDATE (отметьте два варианта)?

- ● **50%** Константы
- ○ **-100%** Переменные
- ● **50%** Выражения от значений полей текущей записи

Функциональные зависимости и поиск данных/Оптимизация выполнения запросов SQL

235. Метод оптимизации выполнения запросов, основанный на синтаксисе:

- ○ При использовании этого метода оптимизатор не рандомно выбирает оптимальный план выполнения запроса
- ○ При использовании этого метода оптимизатор начинает начинает работу без проверки оптимального плана выполнения запроса
- ● При использовании этого метода план составляется на основании существующих путей доступа и их рангов. Все пути доступа ранжируются на основании знаний о правилах и последовательности осуществления этих путей
- ○ Нет правильного ответа

236. Если оптимизатор основывается только на информации о механизмах реализации путей доступа, то метод оптимизации основан на:

- ○ Основан на стоимости
- ● Основан на синтаксисе
- ○ Основан на синтаксисе и стоимости
- ○ Нет правил иного ответа

237. Укажите ранги путей доступа для СУБД основанный на синтаксисе (отметьте три варианта):

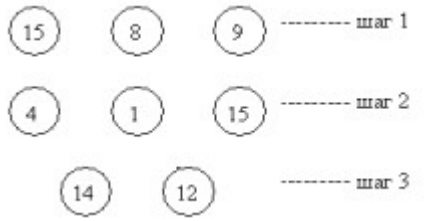
- ● **33%** MAX и MIN по индексированному столбцу
- ● **33%** ORDER BY по индексированному столбцу
- ● **34%** Полный просмотр таблицы
- ○ **-100%** Просмотр таблицы с конца
- ○ **-100%** Просмотр таблицы с середины

238. Укажите ранги путей доступа для СУБД основанный на синтаксисе (отметьте три варианта):

- ● **33%** Ключ индексного кластера
- ● **33%** Составной индекс

- **34%**ORDER BY по индексированному столбцу
- **-100%**Неполный просмотр таблицы
- **-100%**Нет правильного ответа

239. Постройте план выполнения запроса реализованном методом оптимизации по стоимости:



- 15, 4, 14
- 8, 1, 12
- 9, 15, 12
- 8, 1, 14
- Нет правильного варианта

240. Укажите номер в котором описывается метод оптимизации, основанный на стоимости:

- При использовании этого метода оптимизатор сначала строит несколько возможных планов выполнения запроса. Для выбора наиболее перспективных планов оптимизатор в самом начале отбрасывает неэффективные планы и не рассматривает их
- При использовании этого метода оптимизатор строит один возможный план выполнения запроса
- При использовании этого метода оптимизатор строит несколько возможных планов, для выбора более перспективных планов оптимизатор в самом начале отбрасывает эфективные планы
- Нет правильного ответа

241. Если оптимизатор основывается только на информации о механизмах реализации путей доступа , и помимо этого используется статистическая информация о распределении данных, то это метод оптимизации, основанный на:

- Основан на синтаксисе и стоимости
- Основан на стоимости
- Основан на синтаксисе
- Нет правильного ответа

242. Расположите поэтапно оптимизацию запросов в реляционных СУБД:
1)Выбор плана выполнения запроса
2)лексический и синтаксический анализ
3)Выполнение плана
4)Логическая и семантическая оптимизация
5)Генерация процедурного плана выполнения запроса

- 1,2,3,4,5

- 2,4,1,5,3
- 1,3,4,2,5
- 5,3,1,2,4
- 2,4,3,1,5,

243. Укажите к какой фазе, оптимизации выполнения запроса, относятся следующие действия. Лексический анализатор разбивает запрос на лексические единицы – лексемы (наименования полей и таблиц, константы, знаки операций и т.д.). Синтаксический анализатор проверяет синтаксическую правильность запроса.

- Лексический и синтаксический анализ
- Выполнение плана
- Генерация процедурного плана выполнения запроса
- Выбор плана выполнения запроса
- Логическая и семантическая оптимизация

244. Для чего нужно использовать Оптимизация SQL запросов?

- Для улучшения читаемости кода базы данных
- для улучшения интерфейса базы данных
- Для увеличения скорости базы данных
- Нет правильного ответа

Проектирование баз данных/Этапы проектирования баз данных

245. В результате инфологического проектирования БД должна быть создана:

- Инфологическая модель
- Схема БД
- Концептуальная модель
- Реляционная модель данных

246. Концептуальная модель (инфологическая модель) базы данных включает в себя (отметьте два варианта):

- **-100%**Набор схем отношений
- **50%**Описание информационных объектов
- **50%**Описание ограничений целостности
- **-100%**Описание внешних ключей

247. Отметьте 2 преимущества ER моделей:

- **50%**Модели позволяют проектировать базы данных с большим коли- чеством объектов и атрибутов
- **-100%**Атрибуты объектов
- **-100%**Связь

- 50%Наглядность

248. Выберите 2 элемента ER-моделей:

- 50%Связи между объектами
- 50%Атрибуты объектов
- -100%Связь
- -100%Набор схем отношений

249. Связь между сущностями характеризуется (отметьте два варианта):

- -100%Внешними ключами
- 50%Типом связи (1:1, 1:N, N:M)
- 50%Классом принадлежности
- -100%Атрибутами ключей

250. Структура запросов в предметном подходе к проектированию (отметьте два варианта):

- 50%Не определена
- 50%Определена но не полностью
- -100%Определена
- -100%Определена связями

251. Предметный подход к проектированию БД применяется в тех случаях когда (отметьте два варианта):

- 50%Есть представление какую именно информацию они хотели бы хранить в БД
- -100%Нет возможности использования подхода другого вида
- -100%У разработчиков нет чёткого представление о самой ПО
- 50%У разработчиков есть чёткое представление о самой ПО

252. Основное внимание в предметном подходе к проектированию уделяется:

- Составлению схемы данных
- Исследованию ПО
- Проектированию ПО
- Проектированию БД

253. Функциональный подход применяется когда известны (отметьте два варианта):

- 50%Функции некоторой группы лиц
- -100%Схемы БД
- 50%Функции комплекса задач
- -100%Данные для заполнения базы

254. Функциональный подход реализует принцип:

- ○ Проектирования
- ○ "Сущность–связь"
- ○ Логического проектирования БД
- ● "От задач"

Проектирование баз данных/Элементы проектирования баз данных

255. От чего зависит объем вычислительных ресурсов?

- ○ От предполагаемого объема проектируемой базы данных
- ○ От интенсивности использования базы данных
- ○ От использования в многопользовательском или однопользовательском режиме
- ● Все варианты правильные
- ○ Все варианты неправильные

256. Какие критерии являются важнейшими при выборе СУБД (отметьте три варианта):

- ● **33%** Удобство и надежность СУБД в эксплуатации
- ● **33%** Стоимость СУБД и дополнительного программного обеспечения
- ○ **-100%** Наличие графического интерфейса
- ● **34%** Характеристики производительности системы
- ○ **-100%** Характеристики производительности видео- и аудиоадаптера

257. Какой выбор принципиально влияет на весь процесс проектирования БД?

- ● Выбор ядра СУБД
- ○ Выбор интерфейса
- ○ Выбор средств резервного хранения данных БД
- ○ Выбор средств защиты БД

258. От чего зависит логическое проектирование БД?

- ○ От метода доступа к компонентам БД
- ○ От производительности системы
- ● От модели данных, поддерживаемой данной СУБД
- ○ От средств защиты СУБД

259. Что является результатом логического проектирования БД?

- ● Концептуальная схема БД
- ○ Отображение структуры хранения БД
- ○ Модель данных БД
- ○ Реализация методов доступа к данным

260. Этап разработки БД, при котором происходит увязка логической структуры БД и физической среды, называется:

- Разработкой структуры данных
- Логическим проектированием
- Физическим проектированием
- Размещением данных
- Привязка базы

261. На какие аспекты в первую очередь необходимо обращать внимание при разработке защиты данных в СУБД (отметьте два варианта)?

- **50%** Защита от сбоев
- **-100%** Защита от неправильного ввода данных
- **50%** Защита от несанкционированного доступа
- **-100%** Защита от редактирования

262. Какая стратегия применяется для защиты данных от сбоев?

- Система паролей
- Резервное копирование
- Защита от редактирования
- Ограничение доступа к данным

263. Как называются средства автоматизации проектирования?

- FPR-средства
- CASE-средства
- AVPRO-средства
- MS-LOGIC

264. Как дословно переводится аббревиатура CASE-средств для автоматизации проектирования БД?

- Классическая автоматизация сервиса
- Разработка программного обеспечения с помощью компьютера
- Компьютерная автоматическая структура
- Структурная схема инженерии

Проектирование баз данных/Проектирование таблиц

265. Строка в отношении называется:

- Атрибут
- Домен
- Мощность отношения
- Кортеж

266. Арностью отношения в реляционной терминологии определяется:

- ● Числом атрибутов
- ○ Числом кортежей
- ○ Числом доменов
- ○ Числом отношений

267. Какой вид взаимосвязей недопустим в реляционных базах данных:

- ○ Один-к-одному
- ○ Один-к-многим
- ● Многие-ко-многим
- ○ Все ответы являются допустимыми

268. При каком из видов взаимосвязи таблицы могут быть без потери сущности объединены?

- ● Один-к-одному
- ○ Один-к-многим
- ○ Многие-ко-многим
- ○ Нет правильного ответа

269. Отношение А, содержащее внешний ключ, связывающий его с другим отношением В, является:

- ○ Дочерним для всех отношений БД
- ○ Родительским для отношения В
- ○ Родительским для всех отношений БД
- ● Дочерним для отношения В

270. Таблица В связью вида «многие-к-одному» соотносится с таблицей А. Какие из указанных утверждений являются правильными для этого случая (отметьте два варианта):

- ● **50%** Таблица А является родительской
- ○ **-100%** Таблица В является родительской
- ○ **-100%** Таблица А является дочерней
- ● **50%** Таблица В является дочерней

271. Создание дополнительных отношений для удаления связей «многие-ко-многим» используется из-за того, что:

- ○ Дополнительное отношение в любом случае приводит связь к виду «один-к-многим»
- ○ Дополнительное отношение сводит множество таблиц к одной единственной
- ● Дополнительное отношение хранит идентификаторы связанных объектов (таблиц)
- ○ Дополнительное отношение изменяет ключи связанных объектов в соответствии с реляционной моделью БД

272. Ключи дополнительного отношения, поддерживающие связанность и целостность базы данных, по отношению к ключам первоначальных таблиц являются:

- ○ Первичными
- ○ Составными
- ● Внешними
- ○ Альтернативными

273. Количество ключевых атрибутов в таблице не должно превышать:

- ○ 3
- ● Количество атрибутов не ограничено
- ○ 30% от числа атрибутов отношения
- ○ 50% от числа атрибутов отношения

274. В состав атрибутов могут быть включены (отметьте три варианта):

- ● **33%** Первичные ключи
- ● **33%** Внешние ключи
- ○ **-100%** Кортежи доменов
- ● **34%** Неключевые значения доменов

Учреждение образования
«Гомельский государственный университет имени Франциска Скорины»

УТВЕРЖДАЮ

Проректор по учебной работе

ГГУ имени Ф. Скорины

_____ И.В. Семченко

_____

(дата утверждения)

Регистрационный № УД-_____/ уч.

МЕТОДЫ ОБРАБОТКИ БОЛЬШИХ МАССИВОВ ДАННЫХ

Учебная программа учреждения высшего образования по учебной дисциплине для специальности 1-45 80 01 Системы и сети инфокоммуникаций

2020

Учебная программа составлена на основе: образовательного стандарта ОСВО 1-45 80 01-2019 и учебного плана по специальности высшего образования второй ступени (магистратура) 1-45 80 01 Системы и сети инфокоммуникаций регистрационный № I 45-2-01/Д-19, № I 45-2-01/З-19 от 09.04.2019 г

СОСТАВИТЕЛЬ:

В.Н. ЛЕВАНЦОВ старший преподаватель кафедры АСОИ

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой автоматизированных систем обработки информации

(протокол № _11_ от _14.04.2020;

Научно-методическим советом Учреждения образования «Гомельский государственный университет имени Франциска Скорины».

(протокол № _6_ от _20.05.2020;

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дисциплина «Методы обработки больших массивов данных» специальности 1-45 80 01 Системы и сети инфокоммуникаций является дисциплиной компонента учреждения высшего образования модуля «Интеллектуальная обработка данных» и изучается магистрантами второго года обучения.

Дисциплина является актуальной в системе подготовки магистрантов и направлена на формирование управленческих компетенций, способности к определению практических целей в организации научных исследований.

Необходимость дисциплины «Методы обработки больших массивов данных» обусловлена требования образовательного стандарта и учебного плана по специальности 1-45 80 01 Системы и сети инфокоммуникаций

ЦЕЛЬ, ЗАДАЧИ, РОЛЬ УЧЕБНОЙ ДИСЦИПЛИНЫ

Целью дисциплины ««Методы обработки больших массивов данных» является овладение основами обработки больших массивов данных.

Задачами дисциплины являются:

– применять профессиональные знания в области информационно-аналитической деятельности;

– сформировать целостное представление о современных проблемах анализа и обработки больших данных, помочь овладеть опытом разработки и анализа концептуальных и теоретических моделей прикладных задач анализа больших данных с применением моделей Data Mining.разрабатывать маркетинговую стратегию организаций;

– планировать и осуществлять мероприятия, направленные на ее реализацию.

В результате изучения дисциплины магистрант должен:

знать:

– методы решения задач обработки и анализа больших данных,;

– возможности высокопроизводительных вычислительных систем;

– технологии распределенных вычислений, методы и модели Data Mining.

уметь:

– разрабатывать и анализировать концептуальные и теоретические модели прикладных задач анализа больших данных;

– использовать и применять углубленные знания в области обработки и анализа больших данных;

– оценивать время и необходимые аппаратные ресурсы для решения задач анализа и обработки данных;

– создавать алгоритмы анализа и обработки большого объема данных с применением моделей Data Mining.

владеть:

– методами количественного анализа и моделирования, теоретического и экспериментального исследования;

– основными методами, способами и средствами получения, хранения, переработки информации, навыками работы с компьютером как средством управления информацией;

– методами и программными средствами обработки деловой информации, способностью взаимодействовать со службами информационных технологий и эффективно использовать корпоративные информационные системы.

Дисциплина компонента учреждения образования «Методы обработки больших массивов данных» изучается магистрантами 2 года обучения (3 семестр) дневной формы обучения и 2 года обучения (3 семестр) заочной формы обучения для специальности: 1-45 80 01 Системы и сети инфокоммуникаций.

Общее количество часов – 126. (4 зачетных единицы)

Дневная форма обучения: аудиторное количество часов – 46; из них: лекционных занятий – 22 (в том числе УСР – 4), практических занятий – 12(в том числе УСР – 10), лабораторных работ – 12(в том числе УСР – 4) .

Форма отчётности – экзамен.

Заочная форма обучения: аудиторное количество часов – 10; из них: лекционных занятий – 6, практических занятий – 2, лабораторных работ – 2.

Форма отчётности – экзамен.

ТРЕБОВАНИЯ К УРОВНЮ ОСВОЕНИЯ СОДЕРЖАНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

В результате изучения учебной дисциплины «Методы обработки больших массивов данных» формируются следующие компетенции:

СК-10 Владеть методами обработки больших объемов данных.

МЕТОДЫ (ТЕХНОЛОГИИ) ОБУЧЕНИЯ

Основными методами (технологиями) обучения являются:

– словесные, наглядные, практические (по источнику изложения учебного материала);

– репродуктивные, объяснительно-иллюстрированные, поисковые, исследовательские, проблемные и др. (по характеру учебно-познавательной деятельности);

– индуктивные и дедуктивные (по логике изложения и восприятия учебного материала);

– изучение теоретического материала дисциплины на лекциях с использованием компьютерных технологий;

– самостоятельное изучение теоретического материала дисциплины с использованием Internet-ресурсов, информационных баз, методических разработок, специальной учебной и научной литературы;

– закрепление теоретического материала при проведении лабораторных работ с использованием учебного и научного оборудования, выполнения проблемно-ориентированных, поисковых, творческих заданий.

ОРГАНИЗАЦИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ МАГИСТРАНТОВ

При изучении учебной дисциплины рекомендуется использовать следующие формы самостоятельной работы:

– проработка конспекта лекций и учебной литературы;

– самостоятельная подготовка к лабораторным и практическим работам;

– изучение материала, вынесенного на самостоятельную проработку;

– самостоятельная работа в виде решения индивидуальных задач в аудитории во время проведения лабораторных занятий под контролем преподавателя;

– самостоятельное решение во внеурочное время контрольных задач, получаемых на лекциях.

ДИАГНОСТИКА КОМПЕТЕНЦИИ МАГИСТРАНТА

Учебным планом специальности в качестве формы итогового контроля по дисциплине «Методы обработки больших массивов данных» предусмотрен экзамен.

Для текущего контроля и самоконтроля знаний и умений студентов по данной дисциплине используется: выполнение лабораторных работ с их защитой.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Тема 1 Введение в большие данные.

Назначение дисциплины. Основные понятия Роль аналитика данных (Data Scientist). Ключевые компетенции аналитика. Отличия BI от Data Science..

Тема 2 Процесс анализа.

Общая схема анализа. Извлечение и визуализация данных. Этапы моделирования. Процесс построения моделей. Формы представления данных, типы и виды данных. Представления наборов данных**.**

Тема 3 Корреляция и регрессионный анализ.

Коэффициент корреляции. Графическое представление. Постановка задачи регрессионного анализа. Линейная регрессия. Метод наименьших квадратов. Их роль в аналитике больших данных.

Тема 4 Задачи классификации и кластеризации.

Ассоциативные правила. Постановка задачи классификации. Постановка задачи кластеризации. Задача построения ассоциативных правил.

Тема 5 Технологии KDD и Data Mining.

Подготовка данных к анализу. Методика извлечения знаний. Data Mining. Мультидисциплинарный характер Data Mining. Причины распространения KDD и Data Mining. Актуальность технологий Data Mining как средств обработки больших объемов информации.

Тема 6 Парадигма Map Reduce.

Реализация Hadoop. Парадигма Map Reduce. Роль Map Reduce в аналитике больших данных.

Тема 7 Программное обеспечение в области анализа данных.

Аналитические платформы: классификация и особенности применения. Языки визуального моделирования. Начало работы. Понятие сценария и узла обработки. Консолидация данных. Трансформация данных. Визуализация данных.

Тема 8 Подготовка данных.

Визуализация данных. Понимание данных. Методы предварительной подготовки данных. Инструменты и методы визуализации данных

Тема 9 Проблема переобучения.

Регуляризация. Нейронные сети. Машина опорных векторов. Разбор алгоритма нейронных сетей. Разбор алгоритма SVM

Тема 10 Основные понятия теории нейронных сетей.

Основные парадигмы нейронных сетей. Многослойный персептрон: класс решаемых задач, архитектура**.**

Тема 11 Определение дерева решений.

Причины популярности и условия применимости. Структура дерева решений. Выбор атрибута разбиения в узле. Алгоритм ID3, критерий выбора атрибута разбиения ID3, пример работы алгоритма. Проблема переобучения, Неизвестные значения атрибутов, алгоритм C4.5.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА (дневная форма обучения)

Номер раздела, темы, занятия	Название раздела, темы, занятия; перечень изучаемых вопросов	Количество аудиторных часов					Кол-во часов УСР	Формы контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	2	3	4	5	6	7	8	9
1	**Введение в большие данные** 1. Назначение дисциплины. 2. Основные понятия. 3 Роль аналитика данных (Data Scientist). 4 Ключевые компетенции аналитика. 5Отличия BI от Data Science..	**2**						
2	**Процесс анализа** 1 Общая схема анализа. 2 Извлечение и визуализация данных. 3 Этапы моделирования.		**2**				**2**	тест

1	2	3	4	5	6	7	8	9
	4 Процесс построения моделей. 5 Формы представления данных, типы и виды данных. 6 Представления наборов данных**.**							
3	**Корреляция и регрессионный анализ** 1 Коэффициент корреляции. 2 Графическое представление. 3 Постановка задачи регрессионного анализа. 4 Линейная регрессия. 5 Метод наименьших квадратов 6. Их роль в аналитике больших данных.	**2**			**4**			отчет по лабораторной работе
4	**Задачи классификации и кластеризации.** 1 Ассоциативные правила. 2 Постановка задачи классификации. 3 Постановка задачи кластеризации. 4 Задача построения ассоциативных правил.		**2**					реферат
5	**Технологии KDD и Data Mining.** 1 Подготовка данных к анализу.	**2**						реферат

1	2	3	4	5	6	7	8	9
	2 Методика извлечения знаний. Data Mining. 3 Мультидисциплинарный характер Data Mining. 4 Причины распространения KDD и Data Mining. 5Актуальность технологий Data Mining как средств обработки больших объемов информации.							
6	**Парадигма Map Reduce.** 1 Реализация Hadoop. 2 Парадигма Map Reduce. 3 Роль Map Reduce в аналитике больших данных.	**2**			**4**			отчет по лабораторной работе
7	**Программное обеспечение в области анализа данных.** 1 Аналитические платформы: классификация и особенности применения. 2 Языки визуального моделирования. 3 Начало работы. 4 Понятие сценария и узла обработки. 5 Консолидация данных. 6 Трансформация данных. 6 Визуализация данных.	**2-**					**2**	реферат

1	2	3	4	5	6	7	8	9
8	**Подготовка данных.** 1 Визуализация данных. 2 Понимание данных. 3 Методы предварительной подготовки данных. 4 Инструменты и методы визуализации данных	**2**					**2**	тест
9	**Проблема переобучения.** **1 Регуляризация.** **2 Нейронные сети.** **3 Машина опорных векторов.** **4 Разбор алгоритма нейронных сетей.** **5 Разбор алгоритма SVM.**	**2**					**2**	реферат
10	**Основные понятия теории нейронных сетей.** 1 Основные парадигмы нейронных сетей. 2 Многослойный персептрон. 3 Классы решаемых задач. **4 Архитектура сетей..**	**2**						отчет по лабораторной работе

1	2	3	4	5	6	7	8	9
11	**Определение дерева решений.** 1 Причины популярности и условия применимости. 2 Структура дерева решений. 3 Выбор атрибута разбиения в узле. 4 Алгоритм ID3, критерий выбора атрибута разбиения ID3, пример работы алгоритма. 5 Проблема переобучения. 6 Неизвестные значения атрибутов, алгоритм C4.5.	**2**	**2**					реферат
	Всего по дисциплине	**18**	**6**		**8**		**8**	**экзамен**

Старший преподаватель кафедры АСОИ В.Н. Леванцов

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА (заочная форма обучения)

Номер раздела, темы, занятия	Название раздела, темы, занятия; перечень изучаемых вопросов	Количество аудиторных часов					Кол-во часов УСР	Формы контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	2	3	4	5	6	7	8	9
1	**Введение в большие данные** 1. Назначение дисциплины. 2. Основные понятия. 3 Роль аналитика данных (Data Scientist). 4 Ключевые компетенции аналитика. 5Отличия BI от Data Science..	**Самостоятельное изучение**						
2	**Процесс анализа** 1 Общая схема анализа. 2 Извлечение и визуализация данных. 3 Этапы моделирования.		**2**					реферат т

1	2	3	4	5	6	7	8	9
	4 Процесс построения моделей. 5 Формы представления данных, типы и виды данных. 6 Представления наборов данных**.**							
3	**Корреляция и регрессионный анализ** 1 Коэффициент корреляции. 2 Графическое представление. 3 Постановка задачи регрессионного анализа. 4 Линейная регрессия. 5 Метод наименьших квадратов 6. Их роль в аналитике больших данных.	**Самостоятельное изучение**						
4	**Задачи классификации и кластеризации.** 1 Ассоциативные правила. 2 Постановка задачи классификации. 3 Постановка задачи кластеризации. 4 Задача построения ассоциативных правил.	**Самостоятельное изучение**						
5	**Технологии KDD и Data Mining.** 1 Подготовка данных к анализу.	**Самостоятельное изучение**						

1	2	3	4	5	6	7	8	9
	2 Методика извлечения знаний. Data Mining. 3 Мультидисциплинарный характер Data Mining. 4 Причины распространения KDD и Data Mining. 5Актуальность технологий Data Mining как средств обработки больших объемов информации.							
6	**Парадигма Map Reduce.** 1 Реализация Hadoop. 2 Парадигма Map Reduce. 3 Роль Map Reduce в аналитике больших данных.				**2**			отчет по лабораторной работе
7	**Программное обеспечение в области анализа данных.** 1 Аналитические платформы: классификация и особенности применения. 2 Языки визуального моделирования. 3 Начало работы. 4 Понятие сценария и узла обработки. 5 Консолидация данных. 6 Трансформация данных. 6 Визуализация данных.	**Самостоятельное изучение**						

1	2	3	4	5	6	7	8	9
8	**Подготовка данных.** 1 Визуализация данных. 2 Понимание данных. 3 Методы предварительной подготовки данных. 4 Инструменты и методы визуализации данных	**Самостоятельное изучение**						
9	**Проблема переобучения.** **6 Регуляризация.** **7 Нейронные сети.** **8 Машина опорных векторов.** **9 Разбор алгоритма нейронных сетей.** **10 Разбор алгоритма SVM.**	**2**						реферат
10	**Основные понятия теории нейронных сетей.** 1 Основные парадигмы нейронных сетей. 2 Многослойный персептрон. 3 Классы решаемых задач. **4 Архитектура сетей..**	**2**						тест

1	2	3	4	5	6	7	8	9
11	**Определение дерева решений.** 1 Причины популярности и условия применимости. 2 Структура дерева решений. 3 Выбор атрибута разбиения в узле. 4 Алгоритм ID3, критерий выбора атрибута разбиения ID3, пример работы алгоритма. 5 Проблема переобучения. 6 Неизвестные значения атрибутов, алгоритм C4.5.	**2**						тест
	Всего по дисциплине	**6**	**2**		**2**			**экзамен**

Старший преподаватель кафедры АСОИ В.Н. Леванцов

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ТЕМ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

1. Процесс анализа .
2. Задачи классификации и кластеризации.
3. Технологии KDD и Data Mining.
4. Программное обеспечение в области анализа данных.
5. Проблема переобучения.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ТЕМ ЛАБОРАТОРНЫХ ЗАНЯТИЙ

1 Корреляция и регрессионный анализ
2 Парадигма Map Reduce.
3 Основные понятия теории нейронных сетей.

ФОРМЫ КОНТРОЛЯ ЗНАНИЙ

1 Отчеты по лабораторным работам.

2 Тестирование.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ НЕОБХОДИМОГО ОБОРУДОВАНИЯ И КОМПЬЮТЕРНЫХ ПРОГРАММ

1 Класс современных персональных ЭВМ.

2 Современные средства разработки программ.

3 Современные офисные пакеты.

РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ И ВЫПОЛНЕНИЮ УСР

Для самостоятельного изучения выделяются следующие темы:
- характеристики информационных потоков в организации;
- управление информационными потоками;
- информационные потоки в системе управления;
- проблема коммуникаций при обработке информационных потоков;
- построение корпоративной информационной системы.

Тема 1 Корреляция и регрессионный анализ – 2 часа

Цели: 1) овладеть знаниями по данной теме, терминологией и методологией; 2) сформировать компетенцию в умении определения характеристик информационных потоков.

Виды заданий УСР по теме с учетом модулей сложности:

А) Задания, формирующие знания по учебному материалу на уровне узнавания:

1. Соотнесите термины с определениями.
2. Исправьте ошибки в определениях.
3. Вставьте в определение соответствующий термин.

Форма выполнения заданий - индивидуальная.

Форма контроля выполнения заданий – тесты, лабораторная работа.

Б) Задания, формирующие компетенции на уровне воспроизведения:

1. Дайте определения терминам.
2. Приведите примеры, подтверждающие или опровергающие правильность утверждений.
3. Объясните принципы древних систем кодирования.

Форма выполнения заданий – индивидуальная.

Форма контроля выполнения заданий – тесты, контрольные вопросы.

В) Задания, формирующие компетенции на уровне применения полученных знаний:

1. Опишите принципы формирования информационных потоков.
2. Приведите классификацию характеристик информационных потоков.
3. Соотнесите названия характеристик информационных потоков с их описанием.

Форма выполнения заданий - индивидуальная.

Форма контроля выполнения заданий – реферат, лабораторная и практическая работа.

Учебно-методическое обеспечение:

1) Рекомендуемая основная и дополнительная литература.

2) Конспект лекций по дисциплине.

3) Информация в сети Интернет.

Тема 2 Технологии KDD и Data Mining. – 2 часа

Цели: 1) овладеть знаниями по данной теме, терминологией и методологией; 2) сформировать компетенцию в управлении информационными потоками в организации.

Виды заданий УСР по теме с учетом модулей сложности:

А) Задания, формирующие задания по учебному материалу на уровне узнавания:

1. Соотнесите термины с определениями.
2. Исправьте ошибки в определениях.
3. Вставьте в определение соответствующий термин.

Форма выполнения заданий - индивидуальная.

Форма контроля выполнения заданий – тесты.

Б) Задания, формирующие компетенции на уровне воспроизведения:

1. Дайте определения терминам.
2. Приведите примеры, подтверждающие или опровергающие правильность утверждений.
3. Опишите принципы применения криптографических систем.

Форма выполнения заданий – индивидуальная.

Форма контроля выполнения заданий – тесты, контрольные вопросы.

В) Задания, формирующие компетенции на уровне применения полученных знаний:

1. Приведите критерии управления информационными потоками.
2. Приведите примеры применения управления информационными потоками.
3. Опишите свойства информационных потоков.

Форма выполнения заданий - индивидуальная.

Форма контроля выполнения заданий – реферат, лабораторная и практическая работа.

Учебно-методическое обеспечение:

1) Рекомендуемая основная и дополнительная литература.
2) Конспект лекций по дисциплине.
3) Информация в сети Интернет.

Тема 3 Парадигма Map Reduce – 2 часа

Цели: 1) овладеть знаниями по данной теме, терминологией и методологией; 2) сформировать компетенцию в применении систем управления.

Виды заданий УСР по теме с учетом модулей сложности:

А) Задания, формирующие задания по учебному материалу на уровне узнавания:

1. Соотнесите термины с определениями.
2. Исправьте ошибки в определениях.
3. Вставьте в определение соответствующий термин.

Форма выполнения заданий - индивидуальная.

Форма контроля выполнения заданий – тесты.

Б) Задания, формирующие компетенции на уровне воспроизведения:

1. Дайте определения терминам.

2. Приведите примеры, подтверждающие или опровергающие правильность утверждений.

3. Объясните принципы работы информационных потоков в системе управления.

Форма выполнения заданий – тесты.

Форма контроля выполнения заданий – тесты, контрольные вопросы.

В) Задания, формирующие компетенции на уровне применения полученных знаний:

1. Опишите принципы работы информационных потоков в системе управления.

2. Приведите примеры информационных потоков в системе управления.

3. Продемонстрируйте принципы использования систем управления.

Форма выполнения заданий - индивидуальная.

Форма контроля выполнения заданий – реферат, лабораторная и практическая работа.

Учебно-методическое обеспечение:

1) Рекомендуемая основная и дополнительная литература.

2) Конспект лекций по дисциплине.

3) Информация в сети Интернет.

Тема 4 Проблема переобучения. – 2 часа

Цели: 1) овладеть знаниями по данной теме, терминологией и методологией; 2) сформировать компетенцию в умении определять проблемы коммуникаций при обработке информационных потоков.

Виды заданий УСР по теме с учетом модулей сложности:

А) Задания, формирующие задания по учебному материалу на уровне узнавания:

1. Соотнесите термины с определениями.

2. Исправьте ошибки в определениях.

3. Вставьте в определение соответствующий термин.

Форма выполнения заданий - индивидуальная.

Форма контроля выполнения заданий – тесты.

Б) Задания, формирующие компетенции на уровне воспроизведения:

1. Дайте определения терминам.

2. Приведите примеры, подтверждающие или опровергающие правильность утверждений.

3. Объясните принципы определения проблем коммуникаций.

Форма выполнения заданий – тесты.

Форма контроля выполнения заданий – тесты, контрольные вопросы.

В) Задания, формирующие компетенции на уровне применения полученных знаний:

1. Опишите принципы определения проблем коммуникаций при обработке информационных потоков.

2. Приведите примеры определения проблем коммуникаций при обработке информационных потоков.

3. Продемонстрируйте принципы использования обработки информационных потоков.

Форма выполнения заданий - индивидуальная.

Форма контроля выполнения заданий – реферат, практическая работа, лабораторная работа.

Учебно-методическое обеспечение:

1) Рекомендуемая основная и дополнительная литература.

2) Конспект лекций по дисциплине.

3) Информация в сети Интернет.

Тема 5 Определение дерева решений.– 2 часа

Цели: 1) овладеть знаниями по данной теме, терминологией и методологией; 2) сформировать компетенцию в умении построения корпоративной информационной системы.

Виды заданий УСР по теме с учетом модулей сложности:

А) Задания, формирующие задания по учебному материалу на уровне узнавания:

1. Соотнесите термины с определениями.

2. Исправьте ошибки в определениях.

3. Вставьте в определение соответствующий термин.

Форма выполнения заданий - индивидуальная.

Форма контроля выполнения заданий – тесты.

Б) Задания, формирующие компетенции на уровне воспроизведения:

1. Дайте определения терминам.

2. Приведите примеры, подтверждающие или опровергающие правильность утверждений.

3. Объясните принципы построения корпоративной информационной системы.

Форма выполнения заданий – тесты.

Форма контроля выполнения заданий – тесты, контрольные вопросы.

В) Задания, формирующие компетенции на уровне применения полученных знаний:

1. Опишите принципы построения корпоративной информационной системы.

2. Приведите примеры корпоративной информационной системы.

3. Продемонстрируйте принципы построения корпоративной информационной системы.

Форма выполнения заданий - индивидуальная.

Форма контроля выполнения заданий – лабораторная и практическая работа.

Учебно-методическое обеспечение:

1) Рекомендуемая основная и дополнительная литература.

2) Конспект лекций по дисциплине.

3) Информация в сети Интернет.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

ОСНОВНАЯ

1 Лбов, Геннадий Сергеевич. Анализ данных и знаний : учебное пособие / Г.С. Лбов ; Федер. агентство по образованию, Новосиб. гос. ун-т, Мех.-мат. фак .— Новосибирск : Новосибирский государственный университет, 2010 .— 107 с.

2 Ильин, Н.И. Системный подход в управлении строительством./ Н.И. Ильин - М.:Стройиздат, 2004. – 420 с.

3 C. B. B. D. Manyika, “Big Data: The Next Frontier for Innovation, Competition, and Productivity,” McKinsey Global Institute, 2011.

4 Виктор Маер-Шенбергер, Кеннет Кукьер. Большие данные: Революция, которая изменит то, как мы живем, работаем и мыслим. — М.: «Манн, Иванов и Фербер», 2013, 240 с.

5 Грабовый, П.Г. Риски в современном бизнесе./ П.Г. Грабовый -М.: Алане, 1994. – 375 с.

6 Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data // EMC Education Services. 2015. — 432p. — ISBN: 978-1-118-876138.

7 DJ Patil. Building Data Science Teams. O’Reilly. 2011. ISBN: 978-1-449-31623-5

8 Trevor Hastie, Elements of statistical learning, Springer, 2009.

9 Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data // EMC Education Services. 2015. — 432p. — ISBN: 978-1-118-876138.

10 J. Hopcroft, R. Kannan. Foundations of Data Science. 2013. — 412 p.

11 Frontiers in Massive Data Analysis, National Research Council, 2013.

12 J. Adler. R in a Nutshell. Second Edition. O’Reilly Media Inc. 2012. ISBN: 978-1-449-31208-4

ДОПОЛНИТЕЛЬНАЯ

1 Christopher M. Bishop, Pattern recognition and machine learning, Springer, 2006

2 Committee on the Analysis of Massive Data; Committee on Applied and Theoretical Statistics; Board on Mathematical Sciences and Their

Applications; Division on Engineering and Physical Sciences; Frontiers in Massive Data Analysis, National Research Council, 2013

3 C. B. B. D. Manyika, “Big Data: The Next Frontier for Innovation, Competition, and Productivity,” McKinsey Global Institute, 2011. URL:

4 Виктор Маер-Шенбергер, Кеннет Кукьер. Большие данные: Революция, которая изменит то, как мы живем, работаем и мыслим. — 5 М.: «Манн, Иванов и Фербер», 2013, 240 с. ISBN 978-5-91657-936-9

6 Big Data analytics: Future architectures, Skills and roadmaps for the CIO – 2011. – IDC/SAS

ЭЛЕКТРОННЫЕ РЕСУРСЫ

1. Воронцов К.В. Математические методы обучения по прецедентам http://www.machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf

. Математические методы распознавания образов Автор: Л.М. Местецкий (Интернет университет высоких технологий) http://www.intuit.ru/department/graphics/imageproc/4/1.html

3. Онлайн курс Machine learning https://www.coursera.org/course/ml

4. Онлайн курс Big Data Overview https://education.emc.com/academicalliance/elearning/Big_Data_Overview/index.htm

5. Онлайн курс R progamming https://www.coursera.org/course/rprog

6. Онлайн курс Introduction to Data Science https://www.coursera.org/course/datasci

7. Онлайн курс «Введение в аналитику больших массивов данных» http://bit.ly/IntuitBDA.

8. Учебник по статистическому обучению http://statweb.stanford.edu/~tibs/ElemStatLearn/

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ

ПО ИЗУЧАЕМОЙ УЧЕБНОЙ ДИСЦИПЛИНЕ

С ДРУГИМИ ДИСЦИПЛИНАМИ СПЕЦИАЛЬНОСТИ

Название дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы по изучаемой учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)
Протоколы дистанционного управления	Кафедра АСОИ	Без изменений	Рекомендовать к утверждению учебную программу в представленном варианте протокол № ____ от _____
Программирование облачных сервисов	Кафедра АСОИ	Без изменений	Рекомендовать к утверждению учебную программу в представленном варианте протокол № ____ от _____
Коммерциализация результатов научно-исследовательской деятельности	Кафедра АСОИ	Без изменений	Рекомендовать к утверждению учебную программу в представленном варианте протокол № ____ от _____
Технологии сетевого взаимодействия инфокоммуникационных систем	Кафедра АСОИ	Без изменений	Рекомендовать к утверждению учебную программу в представленном варианте протокол № ____ от _____
Методы обработки больших массивов данных	Кафедра АСОИ	Без изменений	Рекомендовать к утверждению учебную программу в

			представленном варианте протокол № ____ от _____

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ ПО ИЗУЧАЕМОЙ УЧЕБНОЙ ДИСЦИПЛИНЕ

на _____/_____ учебный год

№№ пп	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры

АСОИ

(протокол № ____ от _____ 20___ г.)

Заведующий кафедрой АСОИ

к.т.н., доцент _____ А.В. Воруев

УТВЕРЖДАЮ

Декан факультета физики и ИТ УО «ГГУ им. Ф. Скорины»

к.ф.-м.н., доцент _____ Д.Л. Коваленко