

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Тульский государственный университет»

Институт прикладной математики и компьютерных наук  
Кафедра «Прикладная математика и информатика»

Утверждено на заседании кафедры  
«Прикладная математика и информатика»  
14 января 2020 г., протокол № 6

Заведующий кафедрой



В.И. Иванов

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ  
по самостоятельной работе студентов  
по дисциплине (модулю)  
«Базы данных»**

**основной профессиональной образовательной программы  
высшего образования – программы бакалавриата**

по направлению подготовки  
**01.03.03 Механика и математическое моделирование**

с направленностью (профилем)  
**Механика деформируемого твердого тела**

Форма обучения: очная

Идентификационный номер образовательной программы: 010303-01-20

Тула 2020 год

## Разработчик методических указаний

Смирнов О.И., доцент каф. ПМиИ, к.ф.-м.н., доцент

(ФИО, должность, ученая степень, ученое звание)



(подпись)

## **Задание на самостоятельную работу**

Основное задание на самостоятельную работу студента — изучение основных направлений дальнейшего развития баз данных и систем управления базами данных. Необходимо осуществить поиск и изучение периодической литературы, статей, новых монографий, касающихся этих проблем.

### **Форма отчетности**

По данной теме предусмотрены вопросы в экзаменационных билетах.

### **Краткая теоретическая справка**

Современные базы данных являются основой многочисленных информационных систем. Информация, накопленная в них, является чрезвычайно ценным материалом, и в настоящий момент широко распространяются методы обработки баз данных с точки зрения извлечения из них дополнительных знаний, методов, которые связаны с обобщением и различными дополнительными способами обработки данных. Базы данных в данной концепции выступают как хранилища информации, это направление называется «Хранилища данных» (Data Warehouse).

Для работы с «Хранилищами данных» наиболее значимым становится так называемый интеллектуальный анализ данных (ИАД), или data mining, — это процесс выявления значимых корреляций, образцов и тенденций в больших объемах данных. Учитывая высокие темпы роста объемов накопленной в современных хранилищах данных информации, невозможно недооценить роль ИАД. По мнению специалистов Gartner Group, уже в 1998 г. ИАД вошел в десятку важнейших информационных технологий. В последние годы началось активное внедрение технологии ИАД. Ее активно используют как крупные корпорации, так и более мелкие фирмы, которые серьезно относятся к вопросам анализа и прогнозирования своей деятельности. Естественно, на рынке программных продуктов стали появляться соответствующие инструментальные средства.

Особенно широко методы ИАД применяются в бизнес-приложениях аналитиками и руководителями компаний. Для этих категорий пользователей разрабатываются инструментальные средства высокого уровня, позволяющие решать достаточно сложные практические задачи без специальной математической подготовки. Актуальность использования ИАД в бизнесе связана с жесткой конкуренцией, возникшей вследствие перехода от «рынка продавца» к «рынку покупателя». В этих условиях особенно важно качество и обоснованность принимаемых решений, что требует строгого количественного анализа имеющихся данных. При работе с большими объемами накапливаемой информации необходимо постоянно оперативно отслеживать динамику рынка, а это практически невозможно без автоматизации аналитической деятельности.

В бизнес-приложениях наибольший интерес представляет интеграция методов интеллектуального анализа данных с технологией оперативной аналитической обработки данных (On-Line Analytical Processing, OLAP). OLAP использует многомерное представление агрегированных данных для быстрого доступа к важной информации и дальнейшего ее анализа.

Системы OLAP обеспечивают аналитикам и руководителям быстрый последовательный интерактивный доступ к внутренней структуре данных и возможность преобразования исходных данных с тем, чтобы они позволяли отразить структуру системы нужным для пользователя способом. Кроме того, OLAP-системы позволяют просматривать данные и выявлять имеющиеся в них закономерности либо визуально, либо простейшими методами (такими как линейная регрессия), а включение в их арсенал нейросетевых методов обеспечивает существенное расширение аналитических возможностей.

В основе концепции оперативной аналитической обработки (OLAP) лежит многомерное представление данных. Термин OLAP ввел Кодд (E. F. Codd) в 1993 году. В своей статье он рассмотрел недостатки реляционной модели, в первую очередь невозможность «объединять, просматривать и анализировать данные с точки зрения множественности измерений, то есть самым понятным для корпоративных аналитиков способом», и определил общие требования к системам OLAP, расширяющим функциональность реляционных СУБД и включающим многомерный анализ как одну из своих характеристик.

Следует заметить, что Кодд обозначает термином OLAP многомерный способ представления данных исключительно на концептуальном уровне. Используемые им термины — «Многомерное концептуальное представление» («Multidimensional conceptual view»), «Множественные измерения данных» («Multiple data dimensions»), «Сервер OLAP» («OLAP server») - не определяют физического механизма хранения данных (термины «многомерная база данных» и «многомерная СУБД» не встречаются ни разу).

Часто в публикациях аббревиатурой OLAP обозначается не только многомерный взгляд на данные, но и хранение самих данных в многомерной БД, что в принципе неверно.

По Кодду, многомерное концептуальное представление (multi-dimensional conceptual view) является наиболее естественным взглядом управляющего персонала на объект управления. Оно представляет собой множественную перспективу, состоящую из нескольких независимых измерений, вдоль которых могут быть проанализированы определенные совокупности данных. Одновременный анализ по нескольким измерениям данных определяется как многомерный анализ. Каждое измерение включает направления консолидации данных, состоящие из серии последовательных уровней обобщения, где каждый вышестоящий уровень соответствует большей степени агрегации данных по соответствующему измерению. Так, измерение Исполнитель может определяться направлением консолидации, состоящим из уровней обобщения «предприятие—подразделение—отдел-служащий». Измерение Время может даже включать два направления консолидации — «год—квартал—месяц—день» и «неделя—день»,

поскольку счет времени по месяцам и по неделям несовместим. В этом случае становится возможным произвольный выбор желаемого уровня детализации информации по каждому из измерений. Операция спуска (*drilling down*) соответствует движению от высших ступеней консолидации к низшим; напротив, операция подъема (*rolling up*) означает движение от низших уровней к высшим.

Следующим новым направлением в развитии систем управления базами данных является направление, связанное с отказом от нормализации отношений. Во многом нормализация отношений нарушает естественные иерархические связи между объектами, которые достаточно распространены в нашем мире. Возможность сохранять их на концептуальном (но не на физическом) уровне позволяет пользователям более естественно отражать семантику предметной области. В настоящий момент уже существует теоретическое обоснование работы с ненормализованными отношениями и практические реализации подобных систем.

Дальнейшим расширением в структурных преобразованиях являются объектно-ориентированные базы данных. В объектно-ориентированной парадигме предметная область моделируется как множество классов взаимодействующих объектов. Каждый объект характеризуется набором свойств, которые являются как бы его пассивными характеристиками и набором методов работы с этим объектом. Работать с объектом можно только с использованием его методов. Атрибуты объекта могут принимать определенное множество допустимых значений, набор конкретных значений атрибутов объекта определяет его состояние. Используя методы работы с объектом, можно изменять значение его атрибутов и тем самым как бы изменять состояние самого объекта. Множество объектов с одним и тем же набором атрибутов и методов образует класс объектов. Объект должен принадлежать только одному классу (если не учитывать возможности наследования). Допускается наличие примитивных предопределенных классов, объекты-экземпляры которых не имеют атрибутов: целые, строки и т.д. Класс, объекты которого могут служить значениями атрибута объектов другого класса, называется доменом этого атрибута.

Одной из наиболее перспективных черт объектно-ориентированной парадигмы является принцип наследования. Допускается порождение нового класса на основе уже существующего класса, и этот процесс называется наследованием. В этом случае новый класс, называемый подклассом существующего класса (суперкласса), наследует все атрибуты и методы суперкласса. В подклассе, кроме того, могут быть определены дополнительные атрибуты и методы. Различаются случаи простого и множественного наследования. В первом случае подкласс может определяться только на основе одного суперкласса, во втором случае суперклассов может быть несколько. Если в языке или системе поддерживается единичное наследование классов, набор классов образует древовидную иерархию. При поддержании множественного наследования классы связаны в ориентированный граф с корнем, называемый решеткой классов. Объект подкласса считается принадлежащим любому суперклассу этого класса.

Можно считать, что наиболее важным качеством ООБД (объектно-ориентированной базы данных), которое позволяет реализовать объектно-ориентированный подход, является учет поведенческого аспекта объектов.

В прикладных информационных системах, основывавшихся на БД с традиционной организацией (вплоть до тех, которые базировались на семантических моде-

лях данных), существовал принципиальный разрыв между структурной и поведенческой частями. Структурная часть системы поддерживалась всем аппаратом БД, ее можно было моделировать, верифицировать и т.д., а поведенческая часть создавалась изолированно. В частности, отсутствовали формальный аппарат и системная поддержка совместного моделирования и гарантий согласованности структурной (статической) и поведенческой (динамической) частей. В среде ООБД проектирование, разработка и сопровождение прикладной системы становятся процессом, в котором интегрируются структурный и поведенческий аспекты. Конечно, для этого нужны специальные языки, позволяющие определять объекты и создавать на их основе прикладную систему.

Специфика применения объектно-ориентированного подхода для организации и управления БД потребовала уточненного толкования классических концепций и некоторого их расширения.

Прежде всего, возникло направление, которое предполагает возможность хранения объектов внутри реляционной БД, тогда дополнительно необходимо предусмотреть хранение и использование специфических методов работы с этими объектами, а это в свою очередь требует расширения стандарта языка SQL. Частично это уже сделано в новом стандарте SQL3, однако там далеко не все вопросы получили однозначное разрешение.

Однако часть разработчиков придерживается мнения о необходимости полного отказа от реляционной парадигмы и перехода на объектно-ориентированную парадигму. Для перехода к объектно-ориентированным БД стандарт объектного проектирования был дополнен стандартизованными средствами доступа к базам данных (стандарт ODMG93).

Поставщики коммерческих СУБД немедленно отреагировали на эту потребность. Практически каждая уважающая себя фирма обратилась к объектным технологиям и продуктивно сотрудничает с разработчиками объектно-ориентированных СУБД. IBM и Oracle доработали свои СУБД (соответственно, DB2 и ORACLE), добавив объектную надстройку над реляционным ядром системы. Другой путь выбрал Informix, который приобрел серьезную объектно-реляционную СУБД Illustra и встроил ее в свою СУБД. В результате получился продукт, именующийся универсальным сервером. Другой лидер рынка СУБД — Computer Associates, поступил иначе. Он сделал ставку на чисто объектную базу Jasmine, активно пропагандируя ее достоинства. Кто прав — покажет будущее.

Следующим направлением развития баз данных является появление так называемых темпоральных баз данных, то есть баз данных, чувствительных ко

времени. Фактически БД моделирует состояние объектов предметной области в некоторый текущий момент времени. Однако в ряде прикладных областей необходимо исследовать именно изменение состояний объектов во времени. Если использовать чисто реляционную модель, то требуется строить и хранить дополнительно множество отношений, имеющих одинаковые схемы, отличающиеся временем существования или снятия данных. Гораздо перспективнее и удобнее для этого использовать специальные механизмы снятия срезов по времени для определенных объектов БД. Основной тезис темпоральных систем состоит в том, что для любого объекта данных, созданного в момент времени  $t_1$  и уничтоженного в момент времени  $t_2$ , в БД сохраняются (и доступны пользователям) все его состояния во временном интервале  $[t_1, t_2]$ . При обозначении интервала квадратные скобки означают, что граница интервала включена в него, а круглые скобки означают, что точка на временной оси, соответствующая границе интервала, не включается в интервал. И действительно, если объект уничтожен в момент времени  $t_2$ , то в этой точке временной оси он уже не существует, поэтому мы оставляем правую границу временного интервала открытой.

Еще одним из перспективных направлений развития баз данных является направление, связанное с объединением технологии экспертных систем и баз данных и развитие так называемых дедуктивных баз данных. Эти базы основаны на выявлении новых знаний из баз данных не путем запросов или аналитической обработки, а путем использования правил вывода и построения цепочек применения этих правил для вывода ответов на запросы. Для этих баз данных существуют языки запросов, отличные от классического SQL. В экспертных системах также знания экспертов хранятся в форме правил, чаще всего используются так называемые продукционные правила типа «если *описание ситуации*, то *описание действия*». Хранение подобных правил и организация вывода на основании имеющихся фактов под силу современным СУБД.

И наконец, последним, но, может быть, самым значительным направлением развития баз данных является перспектива взаимодействия Web-технологии и баз данных. Простота и доступность Web-технологии, возможность свободной публикации информации в Интернете, так чтобы она была доступна любому количеству пользователей, несомненно, сразу завоевали авторитет у большого числа пользователей. Однако процесс накопления слабоструктурированной информации быстро проходит и далее наступает момент обеспечения эффективного управления этой разнообразной информацией. И это уже серьезная проблема. Некоторые исследователи даже вывели определенную тенденцию, которая выражается в том, что наиболее популярные сайты со временем становятся неуправляемыми, в море информации невозможно отыскать то, что требуется. С одной стороны, Web представляет собой одну громадную базу данных. Однако до сих пор, вместо того чтобы превратиться в неотъемлемую часть инфраструктуры Web, базы данных остаются на вторых ролях. Во-первых, дизайнеры крупнейших Web-серверов с миллионами страниц содержимого постепенно перекладывают задачи управления страницами с файловых систем на

системы баз данных. Во-вторых, системы баз данных используются в качестве серверов электронной коммерции, помогая отслеживать профили, транзакции, счета и инвентарные листы. В-третьих, ведущие Web-издатели примериваются к использованию систем баз данных для хранения информационного наполнения, имеющего сложную природу. Однако в подавляющей части Web-узлов, особенно в тех, которые принадлежат провайдерам и держателям поисковых машин, технология баз данных не применяется. В небольших Web-узлах, как правило, используются статические HTML-страницы, хранящиеся в обычных файловых системах.

В будущем статические HTML-страницы все чаще станут заменять системами управления динамически формируемым содержимым. Уже сейчас, например, торговцы по каталогам не просто преобразуют бумажные каталоги в наборы статических HTML-страниц. Фактически они представляют электронный каталог, позволяющий заказчикам оперативно узнать то, что их интересует, не пролистывая ненужную информацию: например, продает ли поставщик серые джемперы большого размера. Продавцы предлагают клиентам персонализированные манекены, позволяющие увидеть, как будет сидеть на них одежда. Для персонализации требуются весьма сложные модели данных.

HTML расширяется до XML, языка расширяемой разметки, который лучше описывает структурированные данные. К сожалению, XML, похоже, способен породить хаос в системах баз данных. Развивающийся подъязык запросов XML напоминает процедурные языки обработки запросов, превалировавшие 25 лет тому назад. Кроме того, XML стимулирует использование кэшей (наборов) данных на стороне клиента с поддержкой обновлений, что заставляет разработчиков погружаться в трясину проблем распределенных транзакций. К несчастью, значительная часть работ по XML происходит без серьезного участия сообщества исследователей систем баз данных.

Авторы Web-публикаций нуждаются в инструментах для быстрого и экономичного построения хранилищ данных, рассчитанных на сложные приложения. Это, в свою очередь, формирует требования к технологии баз данных для создания, управления, поиска и обеспечения безопасности содержимого Web-узлов.

С другой стороны, универсальность Web-клиента становится весьма привлекательной для разработчиков несложных приложений, которые смогут работать с базами данных. В этом случае не требуется установка каждого клиента, достаточно выслать код доступа и клиент автоматически может уже работать с базой данных, при этом вам все равно, где находится клиент, он может работать как в локальной, так и в глобальной сети, если технология это позволяет. А это весьма удобно, если вы можете с любого рабочего места, имея соответствующий пароль, получить доступ к необходимым данным. Подобные системы называются системами, разработанными по интранет-технологии, то есть технологии, использующей принципы технологий Интернета, но реализованные во внутренней локальной сети.

Для разработки Интернет-приложений, которые связаны с базами данных, широко используются новые средства программирования: это язык PERL, язык PHP (Personal Home Page Tools), язык JavaScript и ряд других. Это действительно грандиозно и, главное, очень интересно, но это уже темы для других книг. Пробуйте и дерзайте, я думаю, познакомившись с базами данных, вы еще не раз с ними столкнетесь в жизни. Я желаю вам успехов и корректных запросов к базам данных. Вы ведь уже знаете: каков вопрос, таков и ответ. Любая база данных может стать вашим помощником или мучителем, это зависит от разработчиков, мне хочется, чтобы для вас они всегда играли только первую роль.

### **Список рекомендуемой литературы**

1. Смирнов, Олег Игоревич. Базы данных: модели и проектирование: учеб. пособие / О.И. Смирнов ; ТулГУ .— Тула : Изд-во ТулГУ, 2007 .— 106 с.
2. Советов, Б.Я. Базы данных. Теория и практика : учебник для вузов / Б.Я. Советов, В.В. Цехановский, В.Д. Чертовской .— М. : Высш. шк., 2005 .— 463с.
3. Кузин, А. В. Базы данных : учеб. пособие для вузов / А. В. Кузин, С. В. Левонисова.— 2-е изд., стер .— М. : Академия, 2008 .— 316 с.
4. Григорьев, Ю.А. Банки данных: Учебник для вузов / Ю.А. Григорьев, Г.И. Ревунков .— М. : Изд-во МГТУ им.Баумана, 2002 .— 320с.
5. Диго, С.М. Базы данных: проектирование и использование: учебник для вузов / С.М. Диго.— М. : Финансы и статистика, 2005 .— 592 с.
6. Малыхина, М.П. Базы данных: основы, проектирование, использование / М.П. Малыхина.— СПб. : БХВ-Петербург, 2004 .— 512 с.
7. Хомоненко, А.Д. Базы данных : учебник для вузов / А.Д. Хомоненко, В.М. Цыганков, М.Г. Мальцев; под ред. А.Д. Хомоненко .— 4-е изд., доп.и перераб.— СПб. : КОРОНА прнт, 2004 .— 736с.
8. Вестник компьютерных и информационных технологий: научно-технический и производственный журнал.— М.: Машиностроение.
9. <http://e-educ.ru/>
10. <http://www.sql.ru/>
11. <http://www.sql-ex.ru/>