МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО «Тверской государственный университет»

Факультет иностранных языков и международной коммуникации

Направление «Лингвистика»

Кафедра английского языка

Реферат

на тему: «Первые базы данных и словари, сформированные на их основе»

по дисциплине: «Информационные технологии в лингвистике»

|  |  |
| --- | --- |
|  | Выполнила:  студентка 11 группы, дневного отделения  Сергеева Мария Анатольевна |
|  |  |
|  | Проверила:  к.ф.н., доцент кафедры английского языка  Масленникова Е. М. |

ТВЕРЬ

2021

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

Также базой данных (БД) называют совокупность взаимосвязанных данных на машинных носителях, предназначенных для использования в интерактивном (диалоговом) режиме доступа и в программных приложениях. Обычно БД создается для хранения и доступа к данным из некоторой предметной области, то есть представляет собой информационную модель класса объектов.

Кроме того, база данных (БД) — это программа, которая позволяет хранить и обрабатывать информацию в структурированном виде.  
БД — это отдельная независимая программа, которая не входит в состав языка программирования. В базе данных можно сохранять любую информацию, чтобы позже получать к ней доступ.

Историю развития баз данных можно разделить на четыре периода.

1. Период становления – начало 60-х - начало 70-х гг.

В 1960-х информация собиралась и хранилась в файлах. Каждый файл содержал определенные сведения и для охвата всей предметной области требовалось несколько файлов. Например, сведения о товарах хранились в одном файле, а сведения о клиентах - в другом. Информация о приобретении определенных товаров определенными клиентами - в третьем. Такая организация данных вносила свои сложности:

· представление данных в каждом файле было различным; необходимо было согласовывать данные в разных файлах для обеспечения непротиворечивости информации;

· необходимо было выбрать какие данные и в каком виде будут фигурировать в таких файлах, как файл приобретений товаров в примере;

· сложность разработки приложений и их обновления при изменении данных.

· Ситуация требовала улучшения и множество специалистов усердно работали над созданием чего-то более удобного в использовании.

В этот период появляется сам термин «база данных» и создается несколько первоначальных систем. Основой появления баз данных явилось предложение конца 50-х годов использовать файлы для хранения исходных данных. Основное требование к таким файловым системам – быть совместно используемым хранилищем данных. В последующем стало очевидным, что совместно использу­емые данные, должны обладать специфическими свойствами, в частности: независимость данных, отсутствие дублирования и противоречивости, контроль прав доступа к данным, эффективная техника доступа к данным, а также многие другие.

Осознание этих фактов, а также появление больших компьютеров с магнитными дисками в качестве носителей данных привело к появлению в середине 60-х гг. первых систем управления базами данных (СУБД), из которых наиболее развитой оказалась система IMS фирмы IMB, которая поддерживала иерархическую структуру данных. Бахман в 1963 г. разработал первую промышленную систему баз данных IDS. Система IDS поддерживала сетевую организацию данных на магнитных носителях.

Система управления базой данных (СУБД) – это языковые и программные средства для организации, пополнения, модификации и использования БД. Различают универсальные и специализированные СУБД. Универсальные СУБД являются системами широкого профиля и не имеют четко очерченных границ применения, а специализированные создаются для БД конкретного назначения: банковских, бухгалтерских и т. д. Специализированные СУБД в наибольшей степени учитывают специфику предметной области, что отражается в интерфейсе и процедурах обработки информации.

Ассоциация CODASYL, являющаяся органом, разработавшим язык программирования Кобол, в 1967 г. организо­вала рабочую группу по базам данных. Эта группа обобщила языковые спецификации систем баз данных и в 1969 и 1971 гг. издала соответствующие отчеты, которые по наименованию рабочей группы (Data Base Task Group) получили названиеDBTG69,DBTG71. Основой избранного Рабочей группой подхода послужила сетевая структура данных и способы навигации по ней, разработанные в системе IDS, однако сетевая модель данных в отчетах DBTG получила существенное развитие и обоснование.

В этот же период четко выкристаллизовались два подхода относительно проблемы замкнутости систем баз данных. Системы замкнутоготипа характеризуются тем, что они не содержат в своем составе традиционных языков программирования, а имеют непроцедурные языки запросов. Основной целью в данном случае является создание системы, с применением которой мог бы справиться не специалист по программированию К таким системам относились TDMS иUL/1.

Системы с включаемыми языками помимо собственно языков манипулирования базами данных предоставляют также языковые и инструментальные средства разработки приложений, с использованием существующих языков программирования. Этот принцип, в частности, исповедовался DBTG.

В конце данного периода появился термин информационно-управляющая система (MIS). В то время под MIS понималась система баз данных, ориентированная на поиск данных и обеспечивающая возможность работы с удаленного терминала.

Первый этап развития СУБД связан с организацией баз данных на больших машинах типа IBM 360/370, ЕС-ЭВМ и мини-ЭВМ типа PDP11 (фирмы Digital Equipment Corporation — DEC), разных моделях HP (фирмы Hewlett Packard).

Базы данных хранились во внешней памяти центральной ЭВМ, пользователями этих баз данных были задачи, запускаемые в основном в пакетном режиме. Интерактивный режим доступа обеспечивался с помощью консольных терминалов, которые не обладали собственными вычислительными ресурсами (процессором, внешней памятью) и служили только устройствами ввода-вывода для центральной ЭВМ. Программы доступа к БД писались на различных языках и запускались как обычные числовые программы. Мощные операционные системы обеспечивали возможность условно параллельного выполнения всего множества задач. Эти системы можно было отнести к системам распределенного доступа, потому что база данных была централизованной, хранилась на устройствах внешней памяти одной центральной ЭВМ, а доступ к ней поддерживался от многих пользователей-задач.

Особенности этого этапа развития выражаются в следующем:

* Функции управления распределением ресурсов в основном осуществляются операционной системой (ОС).
* Поддерживаются языки низкого уровня манипулирования данными, ориентированные на навигационные методы доступа к данным.
* Значительная роль отводится администрированию данных.
* Проводятся серьезные работы по обоснованию и формализации реляционной модели данных, и была создана первая система (System R), реализующая идеологию реляционной модели данных.
* Проводятся теоретические работы по оптимизации запросов и управлению распределенным доступом к централизованной БД, было введено понятие транзакции.
* Результаты научных исследований открыто обсуждаются в печати, идет мощный поток общедоступных публикаций, касающихся всех аспектов теории и практики баз данных, и результаты теоретических исследований активно внедряются в коммерческие СУБД.

Появляются первые языки высокого уровня для работы с реляционной моделью данных. Однако отсутствуют стандарты для этих первых языков.

2.Период развития – 70-е годы. Концепция баз данных широко распространяется благодаря повышению характеристик аппаратного обеспечения компьютеров. Идет успешное внедрение систем, поддерживающих иерар­хи­чес­кую и сетевую структуры данных.

В 1970 году Э. Ф. Кодд опубликовал статью, которая послужила основой для создания реляционной модели данных. Преимущество такой модели хранения данных заключается в минимальном дублировании данных и исключении некоторых типов ошибок, свойственных другим моделям. Согласно этой модели, данные хранятся в виде таблиц со столбцами и строками. Не все виды таблиц приемлемы для реляционной модели и нежелательные таблицы могут быть нормализованы для удовлетворения требованиям реляционной модели. В процессе нормализации таблица как-правило разбиваются на две или несколько более приемлемых таблиц.

Все этот период продолжалась работа DBTG CODASYL. Была специфицирована система языков для баз данных CODASYL.

Этот период в большей мере характеризуется появлением реляционной модели данных, предложенной в 1970 г. сотрудником института фирмы ИБМ в Сан-Хосе Э. Ф. Коддом, всесторонними исследованиями теоретичес­ких и прикладных вопросов этой модели, разработкой экспериментальных реляционных СУБД. Теоретические исследования привели, в конце концов, к созданию формальной теории баз данных, которая до этого носила описательный характер. На протяжении многих лет многие ведущие формы проводили эксперимента­ль­ные исследования по созданию прототипов реляционных СУБД, повышению их эффективности и функциональ­ности. В конце 70-х гг. появляются первые промышленные реляционные СУБД.

В 1979 году небольшая компания Ashton-Tate выпустила продукт для микрокомпьютеров под названием dBase-II, назвав его реляционной СУБД. Благодаря успешной тактике, компании удалось распространить более 100 000 копий продукта среди пользователей компьютеров Osborne.

3. Период зрелости – 80-е годы. Реляционная модель получила полное теоретическое обоснование. Разработаны крупные реляционные СУБД Oracle, Informix, и другие. Промышленные реляционные системы получают широкое распрост­ране­ние во всех сферах человеческой деятельности. Реляционные системы практически вытеснили с мирового рынка ранние СУБД иерархического и сетевого типа.

Дальнейшее развитие реляционных СУБД шло в следующих направлениях:

Удобство применения. Появление персональных компьютеров сделал принципиальным вопрос удобства использо­ва­ния программ, что также относилось и к СУБД. На протяжении всего этого периода интенсивно развивается внешний интерфейс взаимодействия пользователей с базами данных.

Многоплановость. Изначально базы данных разрабатывались для хранения и обработки символьной информации и традиционно использовались в таких сферах, как обработка экономической информации, статистика, банковское дело, системы резервирования, информационные системы различного направления. Появление спроса к базам данных в нетрадиционных сферах их применения, системы автоматизации проектирования, издательское дело и другие, потребовали хранения в базах данных и обработки изображений, звуков, полнотекстовой информации.

Этот период также характеризуется теоретическими и экспериментальными исследованиями в области баз знаний. Разрабатываются многочисленные экспертные системы, использующие базы знаний. В подавляющем большинстве случаев базы знаний разрабатываются на основе реляционных СУБД.

В середине 1980-х годов пользователи начали объединять свои компьютеры в локальные сети, что привело к возникновению клиент-серверной модели, а также модели с совместным использованием файлов. Сеть позволяла совместно использовать дорогие принтеры и дисковые накопители большой емкости. В перспективе же пользователи хотели совместного использования их баз данных, что стимулировало развитие многопользовательских приложений баз данных для локальных сетей. Поскольку многопользовательская обработка данных в локальной сети отличается от многопользовательской обработки данных на мейнфрейме наличием нескольких вычислителей, возникали дополнительные сложности по координации действий вычислителей. Так появилась клиент-серверная архитектура обработки данных. Существует и более простая, но менее надежная архитектура, снованная на совместном использовании файлов.

Постреляционный период – с начало 90-х гг. В этот период начались проводиться интенсивные исследования по дедуктивным и объектно-ориентированным базам данных, а также разработка исследовательских прототипов таких систем.

Особое место в развитии проблематики объектно-ориентированных СУБД занимает деятельность группы по управлению объектными базами данных ODMG(Object Data Management Group), - неприбыльным консорциумом производителей объектных баз данных и других организаций, заинтересованных в выработке стандартов по хранению объектов в базах данных. ODMGбыла создана в 1991 г. В 1993 г. группа выпустила свой первый стандарт –ODMG-93. В 1995 г. был опубликован усовершенствованный вариант этого стандарта.

Словарь данных, описанный в Словаре вычислений от IBM (IBM Dictionary of Computing) как «центральное хранилище информации о данных, такой как значение, взаимосвязи с другими данными, их источник, применение и формат.»Термин может иметь одно из близких по смыслу значений, относясь к базам данных и СУБД:

* документ, описывающий базу данных или комплект баз данных
* целый компонент СУБД, необходимый для определения её структуры
* часть подпрограммного ПО, расширяющее или подменяющее встроенные словари данных СУБД

Словарь данных содержит информацию об источниках, форматах и взаимосвязях между данными, их описания, сведения о характере использования и распределении ответственности. Словарь данных можно рассматривать как вспомогательную базу данных, в которой хранится информация об основной базе данных.

Пользователи баз данных и разработчики приложений могут получить выгоду от единого стандартизированного документа словаря данных, который перечисляет организацию, содержимое, соглашения по одной или более баз данных. Это обычно включает в себя имена и описания различных таблиц и полей в каждой базе данных, дополнительные детали такие, как тип и длина каждого элемента данных. Не существует универсального стандарта, описывающего уровень детализации в подобном документе, но есть основное описание метаданных о структуре базы данных, а не о самих данных. Документ словаря данных также может включать в себя дополнительную информацию, описывающую кодирование элементов данных. Одним из преимуществ хорошо спроектированного словаря данных является то, что он помогает упорядочить структуру базы данных или большого комплекса распределенных баз данных

В области создания приложений для баз данных может быть полезным добавление дополнительного программного слоя словаря данных, то есть подпрограммного ПО, который будет взаимодействовать с нижележащим словарем данных СУБД. Такой «высокоуровневый» словарь данных может обеспечить дополнительные возможности и степень гибкости, который обойдет ограничения естественного «низкоуровневого» словаря данных, чье главное назначение заключается в поддержке основных функций СУБД, а не требований обычных приложений. Например, высокоуровневый словарь данных может реализовывать альтернативные ER-модели данных, приспособленных под различные приложения, которые совместно используют распространенные базы данных.[[4]](https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D0%BE%D0%B2%D0%B0%D1%80%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85#cite_note-4) Расширения словаря данных также могут помочь и в области оптимизации запросов в распределенных базах данных.

Платформы, предназначенные для быстрой разработки приложений, иногда содержат в себе высокоуровневые инструменты словаря данных, которые могут существенно снизить значимость программных требований к разработке меню, форм, отчетов и прочих компонентов приложений баз данных, включая сами базы данных. Например, PHPLens содержит библиотеку классов языка PHP для автоматизации создания таблиц, индексов и внешних ключей переносимых на разные базы данных. Другим словарем данных, основанным на PHP, является часть набора полезных программ RADICORE, автоматически создающего программные объекты, скрипты и SQL-код для меню и форм с проверкой данных и комплексными объединениями.Для платформы ASP.NET компания Base One International разработала словарь данных, обеспечивающий кросс-СУБД возможности для автоматизированного создания баз данных, проверки данных, улучшения производительности (кэширование и использование индексов), безопасность приложений, а также дополнительные типы данных.

.В наши дни активно развиваются web-приложения баз данных, а также базы данных с использованием Internet-технологий. Web-приложения баз данных делают данные доступными через обозреватель пользователя, в то время как базы данных с использованием Internet-технологий просто используют клиентские обозреватели и технологии типа XML и DHTML для работы с базой данных, не публикуя данные через Internet.

Позволяет быстро найти нужное слово, часто с учётом морфологии и возможностью поиска словосочетаний (примеров употребления), а также с возможностью изменения направления перевода (например, англо-русскийили русско-английский.

Внутренне устроен как база данных со словарными статьями.

Машиночитаемые словари (Machine-readable dictionary, кратко MRD) используются компьютерными программами для решения различных задач, например, для обработки текстов на естественном языке. Машиночитаемые словари являются разновидностью электронных словарей.

### Создание машиночитаемых словарей

В проекте Leipzig Corpora Collection (LCC) корпуса и одноязычные словари строятся на основе текстов, извлекаемых из сети Интернет. LCC включает около 400 словарей. В качестве затравки для поиска текстов в Интернете в проекте LCC использовали текст Всеобщей декларации прав человека, поскольку Декларация содержит около 2000 общеупотребимых слов и переведена на 370 языков и диалектов. Из 200 Википедий были отобраны тексты для компиляции всего 70 словарей. Тексты не всех Википедий были включены в корпус LCC, поскольку многие вики-проекты начинаются с создания статей-заготовок, содержащих почти одинаковые предложения.

Такой подход трудно реализуем для малоресурсных языков, но проект Crúbadán, собрав данные для более чем 2200 языков, показывает, что автоматический поиск для языков, представленных в сети Интернет малым или единичным числом текстов, также возможен. В дальнейшем эти тексты используются для создания словарей, например, в проекте Crúbadán было собрано более 100 млн валлийских слов и половина валлийских текстов с этими словами были переданы Уэльскому университету для создания валлийского словаря.

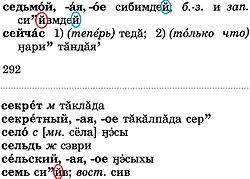
### В работах 1980-х годов предполагали, что на основе машиночитаемых словарей можно построить большие базы знаний. Но в дальнейшем признали, что для этого нужно использовать множество ресурсов, в первую очередь корпусы.

Максимум того, что получилось добиться при извлечении знаний из словарей — это автоматически построить несколько несовершенных таксономий.

Трудности извлечения информации из словарей:

* Преобразование из исходного формата требует больших усилий и эта задача достойна отдельного исследования, однако учёные предпочитают заниматься более научными задачами. Трудность в том, что неоднозначности и противоречия в правилах организации исходного словаря исключают возможность построения полностью автоматического парсера словаря. Построение таких парсеров — задача времязатратная и неблагодарная, поэтому словарей, доступных для компьютерной обработки крайне мало.
* Противоречия и несогласованность в словарях приводят к построению очень разных семантических сетей по разным словарям. Проверка фрагментов пяти главных английских словарей показала, что в 50-70 % случаев информация в толкованиях словарей искажена или отсутствует. Та же плачевная картина получена при анализе трёх главных французских словарей. Из этого следует, что те методы WSD, которые основаны на анализе текстов толкований, в этих многочисленных случаях не будут работать.
* Необходима частичная ручная проверка для построения качественных баз знаний по словаре.
* Необходима интеграция множества источников данных. В комбинировании информации из нескольких словарей есть смысл, поскольку неполнота одного словаря компенсируется другим, имеющим другие лакуны и пропуски информации. В небольшом эксперименте в построении иерархии по пяти английским словарям процент ошибок был снижен с 55-70 % до 5 %. Качество извлекаемой информации при объединении словарей повысилось, но ручная проверка нужна]. С другой стороны, анализ 12 русских словарей показал, что существуют большие размеры пересечений словников словарей. На рисунке представлена доля «уникальной» лексики в каждом из рассмотренных русских словарей.

### Формат машиночитаемых словарей



Фрагмент страницы в русско-ненецком словаре.

Словарные статьи краткие, но включают большой объём разнородных данных: толкование, перевод, словообразование, ударение, грамматические и диалектные пометы

Для использования машиночитаемых словарей их необходимо преобразовать в удобный для работы формат. Формат должен быть достаточно общим для совместимости между разными словарями, для создания единого ПО и повторного использования словарей]. Примером является формат, выработанный сообществом TEI.

С выбором формата словаря тесно связана задача выбора подходящей модели для представления данных машиночитаемого словаря. Если посмотреть любую словарную статью, то можно увидеть, что организация лексикографических данных намного сложнее, чем данные в задаче «товары-поставщик» или организация «базы данных сотрудников». Классические (реляционные) базы данных не являются идеальным решением для лексических баз данных.

В работах, посвящённых исследованиям в области проектирования баз данных, предложена альтернативная модель для лексической базы данных на основе свойств. Особенности этой модели: (1) поддержка вложения значений, (2) есть механизм наследования для исключения избыточной информации.

Словарь данных, описанный в Словаре вычислений от IBM (IBM Dictionary of Computing) как «центральное хранилище информации о данных, такой как значение, взаимосвязи с другими данными, их источник, применение и формат.» Термин может иметь одно из близких по смыслу значений, относясь к базам данных и СУБД:

документ, описывающий базу данных или комплект баз данных

целый компонент СУБД, необходимый для определения её структуры

часть подпрограммного ПО, расширяющее или подменяющее встроенные словари данных СУБД

Словарь данных содержит информацию об источниках, форматах и взаимосвязях между данными, их описания, сведения о характере использования и распределении ответственности. Словарь данных можно рассматривать как вспомогательную базу данных, в которой хранится информация об основной базе данных.

Пользователи баз данных и разработчики приложений могут получить выгоду от единого стандартизированного документа словаря данных, который перечисляет организацию, содержимое, соглашения по одной или более баз данных. Это обычно включает в себя имена и описания различных таблиц и полей в каждой базе данных, дополнительные детали такие, как тип и длина каждого элемента данных. Не существует универсального стандарта, описывающего уровень детализации в подобном документе, но есть основное описание метаданных о структуре базы данных, а не о самих данных. Документ словаря данных также может включать в себя дополнительную информацию, описывающую кодирование элементов данных. Одним из преимуществ хорошо спроектированного словаря данных является то, что он помогает упорядочить структуру базы данных или большого комплекса распределенных баз данных.

В области создания приложений для баз данных может быть полезным добавление дополнительного программного слоя словаря данных, то есть подпрограммного ПО, который будет взаимодействовать с нижележащим словарем данных СУБД. Такой «высокоуровневый» словарь данных может обеспечить дополнительные возможности и степень гибкости, который обойдет ограничения естественного «низкоуровневого» словаря данных, чье главное назначение заключается в поддержке основных функций СУБД, а не требований обычных приложений. Например, высокоуровневый словарь данных может реализовывать альтернативные ER-модели данных, приспособленных под различные приложения, которые совместно используют распространенные базы данных. Расширения словаря данных также могут помочь и в области оптимизации запросов в распределенных базах данных.

Платформы, предназначенные для быстрой разработки приложений, иногда содержат в себе высокоуровневые инструменты словаря данных, которые могут существенно снизить значимость программных требований к разработке меню, форм, отчетов и прочих компонентов приложений баз данных, включая сами базы данных. Например, PHPLens содержит библиотеку классов языка PHP для автоматизации создания таблиц, индексов и внешних ключей переносимых на разные базы данных.Другим словарем данных, основанным на PHP, является часть набора полезных программ RADICORE, автоматически создающего программные объекты, скрипты и SQL-код для меню и форм с проверкой данных и комплексными объединениями. Для платформы ASP.NET компания Base One International разработала словарь данных, обеспечивающий кросс-СУБД возможности для автоматизированного создания баз данных, проверки данных, улучшения производительности (кэширование и использование индексов), безопасность приложений, а также дополнительные типы данных.

Список использованной литературы:

<https://lektsii.org/14-27439.html>

. <https://studfile.net/preview/5376346/page:8/>

<https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D1%8B%D0%B9_%D1%81%D0%BB%D0%BE%D0%B2%D0%B0%D1%80%D1%8C>

<https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85>

<https://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D0%BE%D0%B2%D0%B0%D1%80%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85>