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: «Способы организации баз данных»

По дисциплине «Информационные технологии в лингвистике»

|  |  |
| --- | --- |
|  | Выполнила:студентка группы 14, дневная форма обученияВоловик Полина Максимовна |
|  |  |
|  | Проверила:к.ф.н., доцент, доцент кафедры английского языкаМасленникова Е. М. |

ТВЕРЬ

2021

**Содержание**

|  |  |
| --- | --- |
| База данных ……………………………………………………………… | 2 |
| Модели организации баз данных ………………………………………. | 2 |
| Список используемой литературы …………...………………………… | 10 |

**База данных** — совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.

В наиболее общем виде базу данных определяют как совокупность взаимосвязанной информации. Из этого определения вытекает важная особенность БД, состоящая в том, что БД включает не только сами данные, но и связи между ними. Одной из главных идей базы данных является совместное хранение данных с их описаниями. Благодаря этому хранимые данные становятся «открытыми», понятными для любого числа приложений, работающих с базой. Это делает БД самостоятельным информационным ресурсом, который может многократно использоваться различными приложениями, оставаясь при этом независимым от них.

Кроме того, что база описывает данные, объясняет их значение и структуру, она поддерживает определенные ограничения, накладываемые на эти данные, например определяет тип данных, их размерность и т.п.

Таким образом, *база данных* представляет собой некий информационный ресурс структурированных данных, предназначенный для многоцелевого, многократного использования в конкретных предметных областях.

**Модели организации баз данных**

**Иерархическая модель**

Иерархическая модель базы данных состоит из объектов с указателями от родительских объектов к потомкам, соединяя вместе связанную информацию.

Иерархические базы данных могут быть представлены как дерево, состоящее из объектов различных уровней. Верхний уровень занимает один объект, второй – объекты второго уровня и т. Д.

Между объектами существуют связи, каждый объект может включать в себя несколько объектов более низкого уровня. Такие объекты находятся в отношении предка (объект более близкий к корню) к потомку (объект более низкого уровня), при этом возможно, когда объект-предок не имеет потомков или имеет их несколько, тогда как у объекта-потомка обязательно только один предок. Объекты, имеющие общего предка, называются близнецами.

Иерархической базой данных является, например, файловая система, состоящая из корневой директории, в которой имеется иерархия поддиректорий и файлов.

**Сетевая модель**

Сетевой подход к организации данных является расширением иерархического. В иерархических структурах запись-потомок должна иметь в точности одного предка, в сетевой структуре данных потомок может иметь любое число предков.

В сетевых БД наряду с вертикальными реализованы и горизонтальные связи. Однако унаследованы многие недостатки иерархической и главный из них, необходимость четко определять на физическом уровне связи данных и столь же четко следовать этой структуре связей при запросах к базе.

**Реляционная модель**

СУБД на основе иерархической и сетевой модели не получили широкого распространения на практике.

Реляционная модель данных предложена сотрудником фирмы IBM Эдгаром Коддом и основывается на понятии отношения.

Реляция (от английского relation– отношение) – математический термин из теории множеств, которая легла в основу реляционной модели данных.

В реляционных базах данных вся информация сведена в таблицы, строки которых называются записями, а столбцы называются полями.

Реляционная модель данных некоторой предметной области представляет собой набор отношений, изменяющихся во времени.

Реляционная база данных представляет собой хранилище данных, содержащее набор двухмерных таблиц.

Данные в таблицах должны удовлетворять следующим принципам:

* Значения атрибутов должны быть атомарными (каждое значение, содержащееся на пересечении строки и столбца, должно быть не расчленяемым на несколько значений).
* Значения каждого атрибута должны принадлежать к одному и тому же типу.
* Каждая запись в таблице уникальна.
* Каждое поле имеет уникальное имя.
* Последовательность полей и записей в таблице не существенна.

Данные могут быть организованы по-разному. В зависимости от степени регламентации способа представления данных говорят о неструктурированных и структурированных данных. Между ними находится бесконечное число градаций степени структурированности, но обычно их не различают и используют термин «частично структурированные данные». Например, Microsoft Word.

Во-первых, разные документы могут содержать абсолютно разную информацию, поэтому говорить о знании того, что находится внутри документа, не приходится.

Во-вторых, одни и те же сведения в файле могут быть представлены различными способами: факты могут быть изложены исключительно в виде текста, может быть дано их табличное представление, а может быть приведена диаграмма, иллюстрирующая тот же вопрос. Наконец, информация может быть представлена в комбинированном виде.

Такую информацию называют неструктурированной, она труднее всего поддается автоматической обработке, а при ее анализе требуется человеческий интеллект.

Теперь возьмем простейшую базу данных, например, созданную в настольной системе управления базами данных (СУБД) MS Access. Пусть она состоит из одной таблицы. Информация, которая в пей содержится, имеет жестко заданную структуру:

• определен состав полей записи базы данных (столбцы таблицы);

• каждому полю присвоены определенное имя, тип и свойства;

• все записи базы данных (строки таблицы) имеют одинаковый состав нолей;

• маска ввода, формат вывода, условия проверки полей одинаковы у всех записей.

Все эти сведения хранятся в базе данных вместе с содержимым таблицы, т.е. база данных содержит не только информацию, которая должна быть сохранена, но и метаданные (информацию об информации). Такую информацию называют структурированной, т.е. имеющей известную внутреннюю организацию. Структурированная информация лучше всего поддается автоматизированной обработке.

Представим таблицу Excel, содержащую те же сведения, что и в предыдущем случае. Как и всякая таблица, она имеет более высокую степень структуризации, чем обычный текст. Но в отличие от таблицы Access, здесь меньше ограничений. Так, каждая ячейка таблицы может иметь свой тип, следовательно, в одном столбце могут оказаться сведения разного содержания. Фиксированных имен столбцов нет, а есть их номера или буквенные обозначения, которые не раскрывают содержание хранимой информации. Порядок следования столбцов и строк должен учитываться при обработке, так как обращение к содержимому ячеек осуществляется по их координатам, но не факт, что он соблюдается при эксплуатации. Таким образом, фактическая внутренняя организация данных известна, но не полностью.

Пусть та же информация находится в так называемом плоском файле. В некоторых языках программирования его называют типизированным. Он состоит из записей фиксированной длины, а каждая запись побайтно условно поделена на элементы-поля. Каждое поле, как и в базе данных, имеет свой тип, поэтому запись имеет жесткую структуру. Порядок следования информации в записи типизированного файла неизменен. Но плоский файл не содержит метаданных, поэтому в общем случае неизвестно, какие данные в записи расположены и в каком порядке идут отдельные элементы записи, а также какой тип они имеют. Только если есть документированная информация или тексты исходных файлов программы, можно правильно "разобрать" запись на поля. В противном случае сделать это очень сложно или даже невозможно.

Такую информацию с жесткой структурой при неполных сведениях о ее внутренней организации следует считать частично структурированной.

Необходимо избегать создания и приобретения приложений, которые хранят данные в плоских файлах, так как здесь неизбежно возникает зависимость данных от программ, которые их обрабатывают, а, значит, от фирмы-разработчика или даже от конкретного физического лица. Такие данные сложно сопровождать.

Информация, накапливаемая в базах данных, в отличие от плоских файлов, может жить своей собственной жизнью, независимой от прикладного программного обеспечения. Благодаря наличию метаданных и СУБД структура информации может меняться без утраты содержания накопленных сведений. В противоположность этому любое изменение структуры плоского файла делает программное обеспечение, предназначенное для его обработки, непригодным к использованию, т.е. требует обязательной модификации исходного кода программ, что не всегда возможно и экономически целесообразно.

Возможности по управлению структурированными, неструктурированными и частично структурированными данными разные. Так как наилучшие условия для автоматизированной обработки создаются при наличии структурированных данных, то развитие информационных технологий направлено на то, чтобы максимально возможный объем информации, которая имеется в компании, превратить в структурированную или частично структурированную, что ведет в увеличению доли информации, хранящейся в базах данных и электронных таблицах.

Тем не менее, далеко не вся информация, используемая в компаниях, поддается полной или частичной структуризации. Несмотря на прогресс в ИТ-сфере, сохраняется мощный пласт данных, которые ни при каких обстоятельствах не могут быть структурированы полностью (юридическая информация, договорная, делопроизводственная (входящая и исходящая корреспонденция, приказы, распоряжения, служебные записки, заявления и письма граждан, запросы государственных и правоохранительных органов и т.п.)). Для хранения и поиска такой информации используют электронные образы документов, которые могут храниться как в древовидных каталогах, так и в базах данных с индексацией для полнотекстового поиска или без нее.

Одной из фундаментальных задач управления данными на предприятиях и в организациях является минимизация избыточности данных, когда в базах данных сохраняется много информации одинакового содержания. Так, например, в розничной торговле один товар в течение дня может продаваться тысячи раз, поэтому записи о продажах будут отличаться друг от друга только количеством проданного товара, а такие реквизиты, как наименование товара и его цена будут одинаковыми во всех записях. Таким образом, если не принимать мер, то в базе данных будет многократно указываться, например, длинное наименование товара, что не только увеличит расход памяти на запоминающих устройствах и замедлит обработку информации, но и создаст угрозу возникновения противоречивости данных.

Устоявшийся термин "противоречивость" понимается достаточно широко и под него попадает множество случаев несоответствия данных действительным фактам (разночтения в наименовании товара, названии контрагента, дате осуществления операции, несогласованность сведений с другой информацией, хранимой в базе данных (например, отгрузка товара несуществующему покупателю), осуществленный благодаря ошибкам персонала ввод в текущую БД информации за прошлые или будущие периоды, ввод значений, выходящих за допустимые диапазоны, и т.д.).

Сокращение избыточности в данных происходит за счет нормализации и связывания таблиц, что, в свою очередь, способствует предотвращению противоречивости. Риск возникновения противоречивости уменьшается также за счет контроля со стороны СУБД за целостностью данных, осуществляемого через встроенные механизмы, контроля вводимой информации, что не позволяют нарушить информационные связи между таблицами.

Важной технологической процедурой, характерной для информационных систем, являются транзакции.

***Транзакция*** представляет собой группу операций, которая имеет начало и конец и должна быть выполнена либо полностью, либо не выполнена вообще. При любом исходе транзакции целостность данных не нарушается. Прерывание транзакции (например, из-за сбоя технических средств, отключения питания или программных ошибок) не должно отражаться на состоянии базы данных. Все современные системы управления базами данных работают с транзакциями. Транзакции могут откатываться назад. Благодаря этому возможно корректное исправление ошибок. Термин "транзакция" широко используется при рассмотрении различных аспектов управления данными, так как он фактически означает определенную единицу "информационной работы". Например, бухгалтерская проводка не может считаться завершенной, если запись по дебету счета сделана, а по кредиту – нет.

Информационные системы могут по-разному использовать транзакции. Так, например, платежные системы, работающие с банковскими картами, кроме собственно работы, которую нужно выполнить (перевести деньги со счета на счет) отправляют друг другу так называемые транзакционные данные – информацию о предстоящей работе, позволяющую начать транзакцию по существу или отклонить ее, проанализировав сведения о банковской карте и платежном терминале.

Имеется целый класс систем – OLTP, ориентированных на работу в режиме реального времени с большим числом коротких транзакций, требующих минимальное время отклика. К ним относятся банковские системы, складские системы, торговые системы и др.

Также существуют системы OLAP, приспособленные для обработки сложных аналитических запросов, выполняемых по большому числу связанных таблиц. Такая обработка требует полного просмотра таблиц и потому может требовать значительного времени, а следовательно, создать нагрузку на СУБД и замедлить другие транзакции. Так как аналитические системы не требуют быстрой реакции на запрос, в последние годы нашел признание подход, получивший название OLAP-куба. Его суть состоит в том, что информация из оперативной базы данных извлекается (консолидируется) по нескольким измерениям. После построения куба появляется возможность анализировать информацию оперативно в любых разрезах, которые были предусмотрены при его построении. Построение OLAP-куба требует значительных временных затрат, поэтому выполняется в периоды наименьшей загруженности информационной системы.

**Список используемой литературы:**

1. Научная Электронная Библиотека elibrary.ru – URL: www.elibrary.ru