

Министерство образования Российской Федерации

Ростовский ордена Трудового Красного Знамени
государственный университет

Невская Е.С., Амелина Н.И., Мачулина Л.А.

Введение в системы баз данных

Методические указания для студентов вечернего и дневного отделения
механико-математического факультета

Ростов-на-Дону

2003

Печатается по решению заседания кафедры прикладной математики и программирования от 9 января 2003 г.

АННОТАЦИЯ

Методические указания являются введением в курс «Базы данных», содержат основные понятия и определения баз данных, содержат описания уровней представления данных и типов моделей данных. Основное внимание уделяется реляционной модели данных; дается описание языка данных высокого уровня SQL.

Методические указания предназначены для студентов вечернего и дневного отделения механико-математического факультета РГУ.

Авторы : Е.С.Невская, Н.И.Амелина, Л.А.Мачулина

СОДЕРЖАНИЕ

1. Основные понятия	4
2. База данных	6
3. Уровни представления данных	8
4. Базовые типы моделей данных	10
5. Реляционная модель данных	11
5.1. Структуризация данных	12
5.2. Ограничения целостности	14
5.3. Операции над данными	16
5.4. Язык запросов SQL	19
5.5. Преимущества хранения данных в РБД	26
6. Средства СУБД	27
СЛОВАРЬ ТЕРМИНОВ	28
ЛИТЕРАТУРА	31

1. ОСНОВНЫЕ ПОНЯТИЯ

При наличии большого объема перерабатываемой с помощью компьютера информации возникают задачи обеспечения наилучшего хранения данных, их редактирования и обработки.

Следовательно, нужно

- наилучшим образом *организовать данные*;
- обеспечить наилучшее *управление данными*.

Данные – информация, представленная в определенной форме, пригодной для последующего хранения и обработки.

Предметная область (ПО) – часть реального мира, подлежащая изучению с целью организации управления и последующей автоматизации.

Предметная область определена, если известны существующие в ней объекты, их свойства и связи между ними (объектные отношения).

Объект – это то, о чем необходимо хранить информацию.

Каждый объект характеризуется определенным состоянием, которое описывается с помощью ограниченного набора свойств и связей (отношений) с другими объектами.

Свойства объекта могут не зависеть от его связей с другими объектами, т.е. являются локальными. Если свойства объекта зависят от связей с другими объектами, то они называются реляционными.

Связь между объектами в зависимости от числа входящих в нее объектов характеризуется степенью: $n = 2, 3, \dots, k$.

База данных (БД) – совокупность данных конкретной предметной области, причем данные организованы по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования, и не зависят от программ обработки. В базе данных обеспечивается интеграция логически связанных данных при минимальном дублировании хранимых данных.

Модель данных определяет правила, в соответствии с которыми структурируются данные. Однако структурные правила не обеспечивают возможности полного представления семантики данных и способа их использования. Поэтому должны быть определены операции над данными. Е. Кодд выделяет в модели данных три наиболее существенных компонента:

- совокупность средств определения допустимых структур данных (структуризация данных);
- множество операций, применимых к допустимому состоянию базы данных для поиска или модификации данных;
- множество ограничений целостности, явно или неявно определяющих множество допустимых состояний базы данных.

Структуризация данных определяется разнообразием и количеством типов объектов модели данных, ограничениями на структуру данных. Кроме того, между данными допускается устанавливать дополнительные связи, называемые логическими связями.

Множество операций определяет виды обработок: операции выборки (*поиск*) и операции, изменяющие состояние данных (*редактирование*).

Ограничения целостности поддерживаются средствами, предусмотренными в модели данных для выражения ограничений на значения данных и связи. Эти ограничения характеризуют достоверное состояние данных.

Система управления базами данных (СУБД) – набор программных средств, обеспечивающих создание и обслуживание баз данных и выполнение операций над данными.

Информационная система (ИС) представляет собой систему по сбору, передаче и обработке информации о заданной *предметной области*, снабжающую всех своих пользователей необходимой информацией.

Информационную систему определяют как систему информационных, математических, программных, языковых, организационных и технических средств, предназначенных для централизованного накопления и коллективно-

го многоаспектного использования данных для получения необходимой информации.

Основными компонентами ИС являются:

- собственно *база данных* (БД), содержащая необходимую информацию о предметной области и описание структуры хранимых данных;
- *система управления базой данных* (СУБД), выполняющая типовые процедуры управления данными;
- прикладная программа (*приложение пользователя*), реализующая требуемый алгоритм ведения диалога пользователя с информационной системой для обслуживания БД и решения всего комплекса задач конкретной предметной области.

2. Б А З А Д А Н Н Ы Х

Данные предметной области удобно хранить в интегрированной базе данных по следующей основной причине: такая организация обеспечивает *централизованное управление* данными. Отметим преимущества, которые вытекают из централизованного управления данными.

Сокращается избыточность в хранимых данных ввиду отсутствия дублирования данных в различных объектах модели, что обеспечивает однократный ввод данных и простоту их корректировки.

Устраняются возможности возникновения противоречивости хранимых данных.

Хранимые данные могут использоваться совместно. Совместное использование предполагает не только то, что все данные существующих приложений интегрированы, но также и то, что новые приложения могут быть построены на той же самой базе данных.

Централизованное управление обеспечивает соблюдение стандартов в представлении данных, принятых в данной предметной области.

Могут быть выполнены условия безопасности данных.

Обеспечивается целостность данных. Благодаря централизованному управлению, могут быть определены процедуры проверки, выполняющиеся при операциях запоминания.

Обеспечивается независимость данных. В системе баз данных приложения (программы обработки) не зависят от данных. Это связано с тем, что в разных приложениях одни и те же данные требуется представлять по-разному. Различия могут существовать между представлением данных в приложениях и формой их физического хранения. Кроме того, должна быть возможность изменения структуры хранения или метода доступа без изменения существующих приложений.

Сказанное выше подразумевает, что база данных может развиваться, не оказывая влияния на существующие приложения.

Программное обеспечение, называемое *системой управления базами данных* (СУБД), поддерживает общий интерфейс между всеми пользователями и интегрированной базой данных. СУБД способствует обеспечению секретности и целостности данных, а также позволяет осуществлять контроль данных. Интерфейс СУБД с различными классами пользователей обычно обеспечивается широким диапазоном языков.

Администратор базы данных (АБД) – это лицо (или группа лиц), ответственное за общее управление базой данных. Администратор базы данных является ответственным за анализ потребностей пользователей, проектирование базы данных, её внедрение, обновление, реорганизацию, а также за консультацию и обучение пользователей. Естественно, что администратору базы данных потребуются программные средства для выполнения этой работы, к которым относятся программы загрузки для создания первоначальной версии базы данных, программы реорганизации, программы восстановления.

Важным инструментом администратора базы данных является *словарь данных*, содержащий «данные о данных». В словаре хранятся все схемы, про-

цедуры проверки достоверности данных, проверки полномочий, а также может храниться информация о перекрестных ссылках, показывая связь программ с данными.

3. УРОВНИ ПРЕДСТАВЛЕНИЯ ДАННЫХ

В настоящее время признаны три уровня абстракции для определения структуры базы данных: внутренний, концептуальный и внешний. Внутренний уровень близок к физической памяти, так как связан со способом физического хранения данных (с позиций системного программиста). Внешний уровень отражает представления прикладного программиста или конечного пользователя, так как связан с тем, как отдельные пользователи представляют эти данные. Концептуальный уровень можно представить себе определяющим обобщенное представление пользователей.

Внутренняя модель есть представление самого низкого уровня всей базы данных. Основными компонентами физической базы данных являются физические блоки, хранимые записи, указатели, данные переполнения и промежутки между блоками. Взаимосвязи между хранимыми записями, возникающие в результате их группирования или использования индексных структур, являются частью физической структуры.

Отдельного пользователя интересует только некоторая часть всей базы данных. Представление пользователя об этой части базы данных будет до некоторой степени абстрактным по сравнению с тем, как данные хранятся на физическом уровне. Другими словами можно сказать, что пользователь представляет базу данных посредством внешней модели.

Внешняя модель является информационным содержанием базы данных в том виде, в каком его представляет конкретный пользователь. Внешняя модель определяется посредством *внешней схемы*, которая состоит из описаний всех типов внешних записей этой внешней модели. Определим представление данных на внешнем уровне как представление на уровне реализации.

Представление реализации состоит из логических записей, составляющих их элементов данных и взаимосвязей записей. Наиболее часто применяются три модели данных: иерархическая сетевая и реляционная.

С концептуальным уровнем представления данных согласуется понятие концептуальной модели. *Концептуальная модель* есть представление полного информационного содержания базы данных в абстрактной форме по сравнению со способом физического хранения данных. Это представление может полностью отличаться от представления данных отдельным пользователем. Другими словами, это представление данных скорее ближе к данным в том виде, как они есть, чем в том, какими их видят пользователи при обработке. Концептуальная модель состоит из множества экземпляров различных типов объектов и связей (концептуальных записей).

Структура данных на концептуальном уровне называется *концептуальной схемой* или информационной структурой, которая включает определения каждого типа концептуальных записей. Она является проблемно-ориентированной и системно-независимой, то есть независимой от конкретной СУБД, операционной системы и аппаратного обеспечения компьютера.

Концептуальная структура (или схема) содержит следующие компоненты:

- элементарные данные предметной области, называемые *объектами*;
- элементарные данные, описывающие объекты, называемые *атрибутами*;
- ассоциации между экземплярами элементарных данных, называемые *связями*.

Различают три типа бинарных связей между экземплярами объектов:

- *один-к-одному* (1 : 1);
- *один-ко-многим* (1 : M);
- *многие-ко-многим* (M : N).

Большинство связей предметной области могут быть представлены бинарными связями. Концептуальная схема должна поддерживать согласованность связей.

4. БАЗОВЫЕ ТИПЫ МОДЕЛЕЙ ДАННЫХ

Основным компонентом системы базы данных является модель данных (концептуальная модель).

Тип данных – это определение объектов, их инвариантных свойств (ограничений) и операций, допустимых над ними.

Схему базы данных можно рассматривать как совокупность типов данных. Ограничения, определенные в схеме, могут использоваться для контроля данных того или иного типа. С этой точки зрения база данных – это совокупность значений данных, представляющих собой реализации типов, специфицированных в схеме.

Модель данных можно охарактеризовать как совокупность категорий типов данных. Типы данных, соответствующие этим категориям, используются для представления атрибутов, типов объектов и связей реального мира. Категория определяется посредством других, базовых для нее категорий типов данных.

Кроме того, в моделировании данных акцент делается на определении таких категорий типов данных, которые могли бы использоваться во многих ситуациях общего характера.

Концепция типов данных полезна в моделировании данных с точки зрения получения абстракций и установления соотношений и связей между данными.

Системы баз данных классифицируются в соответствии с выбранной моделью данных.

Широкое распространение получили три типа моделей данных: иерархическая модель, сетевая модель и реляционная модель данных. Эти типы об-

разуют некоторое «базовое» множество. Каждый из типов определяет соответствующую систему управления базами данных (СУБД).

Выбор модели данных зависит от объема информации, сложности решаемых задач и имеющегося технического и программного обеспечения.

Сетевая модель данных базируется на табличных и графовых представлениях. Вершинам графа соответствуют объекты, представленные таблицами, а дугам соответствуют связи между объектами. Ограниченность сетевой модели состоит в невозможности непосредственного представления связей между объектами типа «многие-ко-многим».

Иерархическая модель данных представляется упорядоченным деревом.

Отличие иерархической модели от сетевой состоит в том, что в иерархической модели любой объект может подчиняться только одному объекту вышестоящего уровня, а в сетевой – любой объект может быть подчинен нескольким объектам.

Реляционная модель данных базируется на отношениях и их представлении таблицами. Реляционная модель данных впервые была предложена Е. Коддом. Единственным средством структуризации данных в реляционной модели является *отношение*. Отношения обладают всеми свойствами множеств. Важнейшее свойство языков данных реляционной модели – возможность определять новые отношения, основываясь на существующих отношениях и используя реляционную алгебру или реляционное исчисление.

5. РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

Реляционная модель данных является совокупностью простейших двумерных таблиц – *отношений* (объектов модели).

В теории множеств таблице соответствует термин *отношение* (*relation*), который дал название модели.

5.1. Структуризация данных

Таблица (table) является основным типом структуризации данных (*объектом*) реляционной модели. Структура таблицы определяется совокупностью *столбцов*. В каждой строке таблицы содержится по *одному* значению в соответствующем *столбце*. Такая таблица называется *нормализованной*. В таблице не может быть двух одинаковых строк. Общее число строк не ограничено.

Строки таблицы представляют экземпляры объекта и называются *записями* (records). Столбцы таблицы представляют атрибуты объекта и называются *полями* (fields).

В моделировании данных вводится понятие *домена* (однородного множества). Домены можно рассматривать как множества, из которых черпаются значения семантически значимых объектов и их свойств.

Именованные домены, представляющие семантически значимые объекты, называются *атрибутами*. Атрибуты, определенные на общем домене, обладают одним и тем же свойством.

Поле таблицы – столбец в прямоугольной таблице. Поле таблицы реализует понятие атрибута в реляционной модели,

Дадим формальное определение отношения.

Пусть даны N множеств D_1, D_2, \dots, D_N . Тогда R есть *отношение* над этими множествами, если R есть множество упорядоченных n -местных кортежей вида $\{d_1, d_2, \dots, d_n\}$, где d_1 – элемент из D_1 , d_2 – элемент из D_2 , ..., d_n – элемент из D_N . D_1, D_2, \dots, D_N называются доменами отношения R .

Отношение R можно рассматривать как подмножество декартова произведения множеств D_1, D_2, \dots, D_N .

Число доменов называется *степенью* или размерностью отношения. Текущее число кортежей в отношении называется *мощностью* или кардинальным числом.

В качестве примера рассмотрим реляционную модель данных для предметной области Успеваемость студентов.

Будем предполагать, что студент должен сдать экзамены по некоторым дисциплинам – предметам. После сдачи очередного экзамена преподаватель заполняет направление, в котором указывается ФИО студента, название предмета и полученная студентом оценка.

Ограничимся данными, представленными тремя таблицами (отношениями): Студенты, Предметы, Оценки.

Таблица Студенты содержит следующие данные о каждом студенте: номер студента (номер зачетной книжки), фамилию и инициалы, номер группы, специальность, домашний адрес, телефон.

Таблица Предметы содержит номер предмета и название предмета.

Таблица Оценки содержит номер студента, номер предмета и оценку, полученную данным студентом по данному предмету.

Реляционная модель данных описывается *реляционной схемой*, состоящей из одной или нескольких схем отношений. *Схема отношения* задается именем отношения и именем соответствующих атрибутов.

Для описанной предметной области Успеваемость студентов реляционная схема описывается следующими схемами отношений:

Студенты (*Номер-студента, фамилия-имя-отчество, номер-группы, специальность, домашний адрес, телефон*)

Предметы (*Номер-предмета, название предмета*)

Оценки (*Номер-студента, номер-предмета, балл*)

Основным свойством реляционной модели данных является то, что связи между кортежами таблиц представлены значениями данных в столбцах.

Отношения Студенты и Оценки имеют общий домен – *номера студентов*, отношения Предметы и Оценки имеют общий домен – *номера предметов*.

Реляционная база данных описанной предметной области будет определяться таблицами соответствующих отношений и связями между ними.

5.2. Ограничения целостности

Отношение отображают таблицей, каждая строка которой представляет кортеж. Количество столбцов таблицы равно степени отношения, а число строк – его мощности.

Из того факта, что отношение есть множество, следует, что никакие два кортежа не совпадают и что упорядоченность кортежей несущественна. Это ограничение приводит к понятию ключа отношения.

Ключ отношения – это подмножество атрибутов, совокупность значений которых однозначно идентифицирует кортеж.

Каждое отношение имеет, по крайней мере, один ключ (состоящий из всех атрибутов). Это следует из того, что дублирование строк не допускается. Отношение может иметь и несколько ключей, называемых *возможными ключами*. Для того чтобы исключить тривиальные ключи потребуем, чтобы ключ обладал следующими двумя свойствами:

C1) однозначность идентификации (значение ключа однозначно идентифицирует кортеж отношения);

C2) избыточность (никакое подмножество атрибутов ключа не обладает свойством C1).

Один из возможных ключей выбирается в качестве *первичного ключа* отношения. Первичный ключ не разрешается обновлять. Кроме того, первичный ключ не может принимать значения «не определено».

Вторичный ключ – это такой ключ, значения которого могут повторяться в разных строках таблицы. По ним отыскивается группа строк с одинаковым значением вторичного ключа. Вторичный ключ называют ключом поиска.

Связь – это логическое отношение между объектами, выражающее некоторое ограничение или правило. В реляционной модели вводится понятие ре-

ляционной связи (relation) – это связь между записями, основанная на совпадении значений атрибутов, по которым устанавливается связь.

Внешний ключ – атрибут или комбинация атрибутов некоторого объекта, соответствующие первичному ключу другого объекта, связанного с данным. Иначе говоря, внешний ключ одного объекта представляет собой уникальный идентификатор записи в другом объекте, с которым данный объект связан реляционным отношением.

Для логической связи одной таблицы-отношения, например, R1 с другой таблицей-отношением R2 (назовем её главной), таблица R1 должна содержать *внешний ключ*. *Внешним ключом* подчиненной таблицы R1 является *вторичный ключ* этого отношения, который в то же время является *первичным ключом* в главной таблице. Такие одинаково определенные атрибуты в разных отношениях являются средством логической связи между записями разных таблиц и играют роль *ключа связи*.

Схема данных – представление набора всех объектов, входящих в данную модель, и связей между ними.

Целостность данных – это система правил, используемых для поддержания связей между записями в связанных таблицах.

Большинство конкретных реляционных моделей данных предусматривает средства спецификации явных ограничений целостности, накладываемых на отношения. Например, для атрибута можно задать домен сравнимости, применяемый для указания осмысленности сравнения значений.

Поддержка целостности данных – предотвращение некорректного изменения данных в связанных таблицах, нарушающего установленное отношение между ними.

Например, если из таблицы Студенты удаляется запись, соответствующая некоторому студенту, то необходимо предусмотреть удаление всех записей, соответствующих этому же студенту из таблицы Оценки.

Возможны ограничения на область значений атрибута. Более общие ограничения могут быть заданы посредством утверждений. Утверждения – это предикаты, определяющие условия, которым должна удовлетворять база данных при изменении ее состояния.

5.3. Операции над данными

Операции обработки данных включают операции над строками таблиц и операции над таблицами (отношениями).

Операции над строками - это включение (добавление), удаление, обновление (изменение), то есть операции редактирования.

Основными операциями над отношениями реляционной модели данных являются традиционные операции над множествами: объединение, пересечение, разность (вычитание), декартово произведение, а также специальные операции: выбор, проекция, соединение и деление множеств. Для любой из этих операций результат обработки – новая таблица-отношение.

Специальные операции над отношениями

Выбор – операция выполняется над одним отношением R . Для отношения R по заданному условию осуществляется выборка подмножества кортежей. Результирующее отношение имеет ту же структуру, что и R , но число его кортежей будет меньше или равно исходному.

Например, для отношения *Студенты* в реляционной базе данных *Успеваемость студентов* можно в качестве условия выбора задать конкретное значение для атрибута «специальность», тогда результирующее отношение будет содержать только те кортежи (строки) из таблицы *Студенты*, в которых этот атрибут (столбец) будет иметь заданное значение.

Проекция – операция выполняется над одним отношением, например, R . Операция проекции формирует новое отношение (например, RPR) с заданным подмножеством атрибутов исходного отношения R . Оно может содержать меньше кортежей, так как после отбрасывания в исходном отношении R части

атрибутов (при возможном исключении первичного ключа) могут образоваться кортежи, дублирующие друг друга. Дублирующие кортежи из результирующего отношения исключаются.

Например, если выполнить проекцию отношения *Студенты* в реляционной базе данных *Успеваемость студентов* по атрибуту «специальность», тогда результирующее отношение будет содержать только один атрибут «специальность» и количество кортежей в нем будет равно количеству различных значений атрибута «специальность» из таблицы *Студенты*.

Можно выполнить проекцию отношения по двум и более атрибутам.

Например, для отношения *Студенты* можно выполнить проекцию этого отношения по двум атрибутам: «номер-группы» и «специальность».

Соединение выполняется для заданного условия соединения над двумя логически связанными отношениями. Исходные отношения, например, *R1* и *R2* имеют разные структуры, в которых есть одинаковые атрибуты – внешние ключи (ключи связи). Операция соединения формирует новое отношение, структура которого является совокупностью всех атрибутов исходных отношений. Результирующие кортежи формируются объединением каждого кортежа из *R1* с теми кортежами *R2*, для которых выполняется условие. Условием, как правило, является значение внешнего ключа в исходных отношениях.

Например, можно выполнить соединение над двумя отношениями *Студенты* и *Оценки* реляционной базы данных *Успеваемость студентов* по атрибуту «номер-студента»; тогда в результирующем отношении каждая запись из таблицы *Студенты* соединится поочередно с каждой записью из таблицы *Оценки*, имеющей то же самое значение атрибута «номер-студента», что и запись в таблице *Студенты*.

Деление – операция выполняется над двумя отношениями *R1* и *R2*, имеющими в общем случае разные структуры и некоторые одинаковые атрибуты. В результате операции образуется новое отношение, структура которого получается исключением из множества атрибутов отношения *R1* множества

атрибутов отношения R2. Результирующие строки не должны содержать дубликаты.

Перечисленные операции в той или иной мере реализуются в средствах СУБД, обеспечивающих обработку реляционных таблиц.

Язык SQL, реализованный в большинстве СУБД, содержит все перечисленные операции и полный набор операций над строками (включить, удалить, обновить), а также арифметические операции и операции сравнения.

Язык SQL относится к непроцедурным языкам, в основу которых взята конструкция «предложение», а не конструкция «оператор», используемая в алгоритмических процедурных языках. Следует отметить, что в литературе по языку SQL чаще всего используется термин «оператор» или «инструкция».

Операторы (предложения) SQL разбиты на три категории в соответствии с их функциями:

- язык описания данных (ЯОД), который используется для описания схем отношений;
- язык манипулирования данными (ЯМД), который используется для формирования запроса и поиска данных в таблицах и для редактирования данных;
- операторы защиты и управления данными, которые используются для поддержания целостности базы данных; обычно эти операторы рассматриваются как часть ЯОД.

Опишем на ЯОД реляционную схему предметной области Успеваемость студентов:

```
CREATE TABLE S
  (Ns  (NUMERIC(3,0)),
   Fio (CHAR(20)),
   Ngr (NUMERIC(1,0)),
   Spec (CHAR(4)),
   Addr (CHAR(30)),
   Tel  (CHAR(8)))
```

```
CREATE TABLE P
```

```
( Np (NUMERIC(2,0)),
  Nazv(CHAR(20)))
```

```
CREATE TABLE S_P
( Ns (NUMERIC(3,0)),
  Np (NUMERIC(2,0)),
  Ball (NUMERIC(1,0)))
```

5.4. Язык запросов SQL

Язык SQL был разработан в 1970 году в компании IBM.

SQL – это аббревиатура от английских слов Structured Query Language (Структурированный Язык Запросов).

Язык SQL является подязыком данных, который предназначен для взаимодействия с базой данных, то есть язык SQL – это язык запросов.

Запрос – это операция над отношениями, результатом которой также является отношение. *Система запросов* – это формальная система для выражения запросов. Система запросов образует структуру *языков запросов*, то есть специализированных языков программирования, которые используются в системах баз данных для формулировки команд.

Язык запросов должен иметь набор выразительных средств, чтобы отвечать на широкий круг запросов и формулировать их кратко.

Реляционные языки запросов классифицируются следующим образом:

- языки реляционной алгебры;
- языки реляционного исчисления;
- графические языки;
- языки отображения.

Реляционные языки запросов определяются как спецификационные языки, основными свойствами которых являются полнота, точность и понятность. Следует отметить, что «спецификация описывает, ЧТО надо получить, а не КАК получить», то есть спецификация не должна быть процедурной, динамической, а должна быть декларативной (непроцедурной), статической.

Языки реляционной алгебры относятся к процедурным языкам, остальные языки относятся к непроведурным языкам (языкам высокого уровня).

Теоретические основы первых двух классов языков были заложены трудами Кодда. Языки реляционной алгебры основываются на алгебре отношений, языки реляционного исчисления – на логике предикатов первого порядка.

Графические языки предназначены для работы с дисплеем. Пользователь выражает свой запрос специальными терминами на экране дисплея. Примером графического языка является язык QBE (Query by Example), реализованный в большинстве реляционных СУБД. Язык SQL относится к языкам отображений.

Основной операцией в языке SQL является *отображение*, синтаксически представляющее собой блок SELECT – FROM – WHERE (выбрать – из – где).

Приведем синтаксис предложения SELECT:

```
SELECT < список полей таблиц >
FROM < список имен таблиц >
[ WHERE < логическое выражение > ]
```

Ключевые слова (ключевые параметры) SELECT, FROM являются обязательными, параметр WHERE не является обязательным. Будем обсуждать такие параметры по мере их необходимости.

Рассмотрим описание запросов на языке SQL для конкретной предметной области *Успеваемость студентов*.

Простая выборка

Запрос 1. Получить полные сведения обо всех студентах.

```
SELECT *
FROM S
```

Символ * означает, что в результирующем отношении будут присутствовать все атрибуты отношения S.

Запрос 2. Получить все номера сданных предметов.

```
SELECT UNIQUE Np
FROM S_P
```

В SQL не предусматривается исключение одинаковых строк из результата, если пользователь не потребует этого, задав ключевой параметр UNIQUE. Отметим, что операция исключения одинаковых строк может требовать дополнительных затрат.

Выборка по условию

Запрос 3. Получить номера тех студентов, которые имеют по предмету с номером 2 оценку 5.

```
SELECT Ns
FROM S_P
WHERE Np = 2 AND Ball = 5
```

Логическое выражение после параметра WHERE строится так же, как в алгоритмических языках.

Выборка с упорядочением

Запрос 4. Получить номера и фамилии студентов группы с номером, равным 1 (фамилии должны следовать в лексикографическом порядке).

```
SELECT Ns, Fio
FROM S
WHERE Ngr = 1
ORDER BY Fio DESC
```

Параметр DESC предполагает упорядочение по возрастанию значений в столбце, ASC – упорядочение по убыванию значений в столбце.

Выборка с использованием вложенного отображения

Запрос 5. Получить список фамилий тех студентов, которые сдали предмет с номером 2.

```

SELECT Fio
FROM S
WHERE Ns IS IN
  ( SELECT Ns
    FROM S_P
    WHERE Np = 2 )

```

Параметр IS IN предполагает проверку принадлежности множеству, полученному после выполнения внутреннего SELECT– FROM – WHERE.

Параметр IS NOT IN интерпретируется как «не принадлежит множеству».

Выборка с использованием нескольких уровней вложенности

Запрос 6. Получить список фамилий тех студентов, которые сдали предмет с названием «Мат.анализ».

```

SELECT Fio
FROM S
WHERE Ns IS IN
  ( SELECT Ns
    FROM S_P
    WHERE Np IS IN
      ( SELECT Np
        FROM P
        WHERE Nazv='Мат.анализ') )

```

Выборка с использованием вложенного отображения к одной и той же таблице

Запрос 7. Получить номера тех студентов, которые сдали тот же предмет, что и студент с номером 13.

```

SELECT UNIQUE Ns
FROM S_P
WHERE Np IS IN
  ( SELECT Np

```

```
FROM S_P
WHERE Ns = 13 )
```

Выборка с использованием синонима

Запрос 8. Получить номера всех предметов, которые сданы несколькими студентами.

```
SELECT UNIQUE Np
FROM S_P SPr
WHERE Np IS IN
( SELECT Np
FROM S_P
WHERE SPr.Ns <> Ns )
```

Выборка более чем из одной таблицы

Запрос 9. Для каждого номера предмета получить фамилии студентов, которые его сдали.

```
SELECT Np,Fio
FROM S_P , S
WHERE S_P.Ns = S.Ns
```

Это предложение языка SQL аналогично операции соединения. Существенное различие заключается в том, что фактически соединения S_P и S не происходит, а значит, память на соединение не расходуется.

Выборка, включающая сравнение множеств

Запрос 10. Получить список фамилий тех студентов, которые сдали все предметы.

```
SELECT Fio
FROM S
WHERE ( SELECT Np
```

```

FROM S_P
WHERE Ns = S.Ns )
=
( SELECT Np
  FROM P )

```

Выборка, включающая GROUP BY, HAVING и SET

Запрос 11. Получить номера тех студентов, которые сдали все предметы, что и студент с номером 13.

```

SELECT Ns
  FROM S_P
 GROUP BY Ns
 HAVING SET (Np) CONTAINS
           (SELECT Np
            FROM S_P WHERE Ns = 13)

```

Параметр CONTAINS означает «содержит».

Параметр GROUP BY разделяет рассматриваемую таблицу на такие группы, что внутри любой из этих групп все строки содержат одинаковые значения в указанном столбце.

Затем действует фраза HAVING, являющаяся специальной формой фразы WHERE, но относящаяся не к отдельным строкам, а к группам: Предикат во фразе HAVING всегда ссылается на свойства групп и на основе этого предиката группы целиком либо выбираются, либо отбрасываются.

Значением функции SET становится множество значений, содержащихся в определенном столбце (столбцах) внутри заданной группы.

Операции редактирования

Для включения новой строки используется предложение INSERT, например:


```
INSERT INTO P :
    < 11, 'Информатика' >
```

Для удаления строки используется предложение DELETE, например:

```
DELETE S
    WHERE Ns = 13
```

После выполнения этого предложения из таблицы S будет удалена строка с номером студента, равным 13.

После выполнения предложения

```
DELETE P
```

таблица P станет пустой.

Для изменения значения в заданной строке используется предложение UPDATE, например:

```
UPDATE S
    SET Ngr = 2
    WHERE Ns = 13
```

После выполнения этого оператора у студента с номером 13 номер группы станет равным 2.

Библиотечные функции

В языке SQL существуют следующие библиотечные функции:

COUNT – для получения общего количества строк отношения;

SUM – для суммирования значений в столбце отношения;

AVG – для получения среднего арифметического значения в столбце;

MAX – для получения максимального значения в столбце;

MIN – для получения минимального значения в столбце.

Запрос 12. Получить количество студентов, сдавших предмет с номером, равным 5.

```
SELECT COUNT ( Ns )
    FROM S_P
```

WHERE Np = 5

Запрос 13. Получить средний балл по предмету с номером 5.

```
SELECT AVG ( Ball )
FROM S_P
WHERE Np = 5
```

Запрос 14. Получить номера студентов, сдавших больше 5 предметов.

```
SELECT Ns
FROM S_P
GROUP BY Ns
HAVING COUNT ( Np ) > 5
```

5.4. Преимущества хранения данных в РБД

Преимуществами хранения данных в реляционной базе данных являются следующие:

- каждый элемент данных хранится только в одной таблице (*экономия места*);
- внесение изменений упрощается, *уменьшается риск ошибки* (например, в написании фамилий);
- наличие связей между таблицами *ускоряет обработку взаимосвязанной информации*;
- *ошибочные записи* (с некорректными ссылками) должны автоматически *исключаться*.

6. СРЕДСТВА СУБД

Техническим возможностям персональных компьютеров в настоящее время лучше всего соответствуют реляционные СУБД.

Основными средствами СУБД являются:

- средства задания (описания) структуры объектов базы данных;

- средства конструирования экранных форм, предназначенных для ввода данных, просмотра и их обработки в диалоговом режиме;
- средства создания запросов для выборки данных при заданных условиях, а также выполнения операций по их обработке;
- средства создания отчетов для вывода на печать результатов обработки в удобном для пользователя виде;
- языковые средства – макросы, встроенный алгоритмический язык, язык запросов и др., что позволяет реализовать нестандартные алгоритмы обработки данных;
- средства создания приложений пользователя (генераторы приложений, средства создания меню и панелей управления приложениями), позволяющие объединить различные операции работы с базой данных в единый технологический процесс;
- средства защиты и восстановление данных при аварийных ситуациях, аппаратных и программных сбоях, ошибках пользователя;
- защита данных от несанкционированного доступа средствами ограничения доступа для различных пользователей;
- возможность модификации структуры базы данных без повторной загрузки данных;
- обеспечение независимости программ от данных, позволяющей сохранить программы при модификации структуры базы данных;
- наличие языка запросов высокого уровня, который обеспечивает вывод информации по любому запросу и представлению ее в виде отчетных форм.

СЛОВАРЬ ТЕРМИНОВ

Администрирование базы данных – сопровождение базы данных в процессе разработки, эксплуатации и добавления приложений; включает также создание и необходимую реорганизацию базы данных, создание резервных

копий базы данных, назначение пользователям паролей и санкций доступа к базе данных.

Атрибут – информация, описывающая объект и служащая его идентификатором (поименованная характеристика объекта). В записи данных атрибут представлен типом элемента данных.

База данных – совокупность данных, организованная по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными; не зависит от прикладных программ.

Безопасность данных – защита данных от преднамеренного или непреднамеренного нарушения секретности, искажения или разрушения данных.

Внешний ключ – одно или несколько полей в таблице, которые соответствуют первичному ключу другой таблицы.

Внешняя схема – представление данных с позиции прикладного программиста, логическая структура базы данных.

Внутренняя схема – физическая структура базы данных; представление данных с позиции вычислительной системы, как они выглядят на запоминающем устройстве.

Данные – информация, представленная в определенной форме, пригодной для последующего хранения и обработки.

Домен – область определения значений одного столбца отношения.

Запрос – операция над отношениями, результатом которой также является отношение.

Инструкция SQL – предложение на языке структурированных запросов (SQL), представляющее собой запрос для выборки или обработки данных.

Информация – сведения об окружающем мире и протекающих в нем процессах.

Ключ – совокупность атрибутов, значения которых однозначно определяют кортеж в отношении.

Концептуальная схема – высокоуровневое представление администратора базы данных и пользователя о данных, обычно в виде объектов и связей.

Модель данных – представление о предметной области в виде данных и связей между ними.

Независимость данных – возможность изменения структуры базы данных без изменения пользующихся ею прикладных программ.

Объект – нечто, о чем хранится информация.

Ограничения целостности – определяемые моделью данных или задаваемые схемой базы данных ограничения, обеспечивающие внутреннюю непротиворечивость хранимой информации.

Отношение – конечное множество кортежей из допустимых значений атрибутов схемы отношения. Отношение ассоциируется с таблицей, имена атрибутов – с именами столбцов таблицы, кортежи – со строками таблицы.

Отображение – реляционная операция языка SQL.

Первичный ключ – один или несколько столбцов (атрибутов), которые однозначно идентифицируют каждую запись в таблице.

Предметная область – часть реального мира, подлежащая изучению с целью организации управления и автоматизации.

Реляционная база данных (РБД) – совокупность отношений конкретной предметной области. Предполагается, что отношения логически связаны между собой.

Реляционная СУБД – СУБД, базирующаяся на реляционной модели данных.

Реляционные операции – набор операций манипулирования данными, операндами которых являются отношения; результатом любой реляционной операции является также отношение.

Связь – ассоциация между экземплярами примитивных или агрегированных объектов (записей) данных (например, 1 : 1, 1 : M, M : N).

Система управления базами данных – комплекс программных и языковых средств, предназначенных для создания, ведения и использования баз данных.

Словарь данных – каталог всех типов элементов в базе данных, включающий для каждого типа его определение, формат, источник и применение. Широко распространены автоматические словари данных.

Структура данных – способ объединения нескольких элементов данных в один.

Структура хранения – описание способа организации физического хранения данных в системе: указатели, представление знаков, плавающая запятая, блокирование, метод доступа т.д.

Схема данных – графическое или формальное описание логической структуры базы данных.

Схема отношения – совокупность имен атрибутов, определяющих объект.

Схема реляционной базы данных – совокупность схем отношений приложения.

Тип данных – множество операций, характеризующих определенное множество значений.

Транзакция – группа операций над данными, логически рассматриваемая как одна операция и сохраняющая целостность базы данных.

Целостность базы данных – свойство базы данных, при выполнении которого база данных содержит полную и непротиворечивую информацию, необходимую и достаточную для корректного функционирования приложений; это свойство сохраняется при всех манипуляциях с данными.

Целостность данных – система правил, используемых для поддержания связей между записями в связанных таблицах.

Язык SQL – язык структурированных запросов для выборки, изменения и удаления данных из таблиц базы данных.

ЛИТЕРАТУРА

1. Амелина Н.И., Мачулина Л.А. Методические указания по курсовому проектированию по курсу «Базы данных» для студентов механико-математического факультета вечернего и дневного отделения. – Ростов-на-Дону, УПЛ РГУ, 1999.
2. Дейт К. Введение в системы баз данных. – М: Мир, 1980.
3. Евдокимов В.В. и др. Экономическая информатика. Учебник для вузов. – СПб.: Питер, 1997.
4. Карпова Т.С. Базы данных: модели, разработка, реализация. – СПб.: Питер, 2002.
5. Невская Е.С. Базы данных. Часть 1. Методические указания для студентов 3 и 4 курсов механико-математического факультета. – Ростов-на-Дону, УПЛ РГУ, 1996.
6. Невская Е.С. Базы данных. Часть 2. Методические указания для студентов 3 и 4 курсов механико-математического факультета. – Ростов-на-Дону, УПЛ РГУ, 1997.
7. Ульман Дж. Базы данных на Паскале. – М.: Машиностроение, 1985.
8. Цикритзис Д., Лоховски Ф. Модели данных. – М: Финансы и статистика, 1985.
9. Четвериков В.Н., Ревунков Г.И., Самохвалов Э.Н. Базы и банки данных. – М.: Высш. шк., 1987.