

2.4. Операции

Операция над данными переводит БД из одного состояния в другое

Действия:

- Установка индикаторов текущих элементов (SET CURRENCY)
- Включение или добавление в БД новых элементов (INSERT)
- Обновление или модификация существующих элементов в БД (UPDATE)
- Удаление существующих элементов из БД (DELETE)
- Выборка информации из БД (SELECT)

Способы селекции данных:

- Селекция по указателям текущих (GET NEXT СТУДЕНТ)
- Селекция по данным (GET СТУДЕНТ WHERE Пол = 'ж')
- Селекция по связям (GET СТУДЕНТ WHERE СТУДЕНТ. Фамилия = АВТОМОБИЛЬ. Фамилия владельца)
- Комбинированный способ (GET NEXT СТУДЕНТ WHERE Пол = 'ж' AND СТУДЕНТ. Фамилия = АВТОМОБИЛЬ. Фамилия владельца)

Модель данных мы ранее определили состоящей из множества правил порождения схем и множества операций над данными. Многие считают, что правила применяются к спецификации статических свойств данных, а их динамика выражается с помощью операций.

Не все операции приводят к изменениям в экстенсionaле БД (например, извлечение информации из БД его не меняет). Тем не менее, динамику БД (точнее, системы БД) можно связать и с такими операциями, введя некоторые дополнительные объекты – индикаторы текущих и другие управляющие элементы. Эти объекты в строгом смысле не относятся к объектам БД, но они связаны с ней и могут изменяться в результате выполнения операций. Совместно с конкретной реализацией данных они определяют состояние БД.

Динамические аспекты данных отражаются изменением состояния БД. Рассмотрим, например, последовательную выборку строк таблицы с использованием команды «дать следующую строку». Текущее состояние БД определяется совокупностью значений хранящихся данных, а также значением индикатора текущей, указывающим на строку, к которой осуществлялось последнее обращение. Выполнение операции «дать следующую строку» не приведет к изменению реализации БД, но состояние БД изменится, так как изменится значение индикатора текущей.

Определение 2.4.1. Операции над данными, выражаемые средствами языка манипулирования данными, переводят БД из состояния DBS_i в состояние DBS_{i+1} (или в неопределенное состояние), причем изменение состояния происходит или при изменении реализации данных, или при изменении управляющих элементов.

Каждое состояние БД должно соответствовать ее схеме. Это означает, что выполнение операций не должно приводить к нарушению свойств данных, отраженных в схеме.

Одним из важных моментов определения операций в моделировании данных является обеспечение достаточной простоты их усвоения пользователем.

Определение 2.4.2. Операции обычно задаются в терминах селекции и действия. **Действие** определяет характер операции, а **селекция** – критерий отбора данных, над которыми должно быть произведено действие.

Вполне естественно потребовать, чтобы объектом операции была ограниченная часть БД. Это требование, с одной стороны, исходит из соображений удобства

пользователей, а с другой – из целесообразности ограничения числа обращений к внешней памяти, выполняемых при обработке каждого конкретного запроса к БД. Указанное ограничение достигается селекцией данных. Селекция должна осуществляться вне зависимости от того, какая конкретная операция реализуется. Обычно вопросы селекции данных ассоциируются с доступом к данным. Причина этого кроется в архитектуре ЭВМ, в соответствии с которой выполнению операций должен предшествовать перенос данных в оперативную память.

Определение 2.4.3. В технологии БД выделяют пять основных видов действий над данными:

- а) **установка текущих** – типичные ключевые слова – *SET CURRENCY, FIND*;
- б) **включение** (добавление новых элементов данных в БД) – *INSERT, ADD*;
- в) **обновление** (модификация существующих элементов данных в БД) – *UPDATE, MODIFY*;
- г) **удаление** (исключение элементов данных из БД) – *DELETE, REMOVE*;
- д) **выборка** (получение данных из БД) – *SELECT, GET*.

В зависимости от структурных элементов модели данных указанные действия могут применяться к множествам, отношениям, типам объектов и связей. Таким образом, в конкретной модели элементами данных, над которыми выполняются действия, могут быть элементы множества, кортежи отношения, знаки-объекты и знаки-связи.

Четыре из указанных действий (включение, обновление, удаление и выборка) необходимы в языке манипулирования данными любой СУБД. Каждое из них является ключевым для технологии БД. Без включения невозможно вообще создать экстенционал БД, обновление и удаление позволяют отразить в БД динамику ПрО, а выборка решает основную задачу систем БД – обеспечение пользователей информацией о ПрО. И лишь первое действие (установки текущих) не является необходимым и встречается отнюдь не во всех языках СУБД. Подробнее об этих языках мы расскажем немного позже.

При конкретной структуре и ограничениях целостности результат действий не всегда допустим, действие может нарушать ОЦ. Как правило, в этом случае попытка выполнения действия отвергается, система временно переходит в неопределенное состояние (упоминавшееся в определении 2.4.1), о чем пользователю или программе, инициировавшей операцию, передается сообщение (включающее текст или код ошибки), после чего система возвращает БД в то состояние, которое она имела до начала выполнения ошибочной операции. Кстати, сообщения об ошибках также относятся к управляющим элементам, входящим в расширенное состояние БД.

Определение 2.4.4. Селекция может осуществляться посредством:

- а) логической позиции в БД – **селекция по текущей**;
- б) значений данных – **селекция по данным**;
- в) связей между данными – **селекция по связям**.

Данные можно селектировать, базируясь на их логических позициях в множествах, отношениях и т.п. Хотя модель может и не поддерживать понятие упорядоченности данных, компьютерная реализация в конечном итоге определяет некоторый порядок. Такой порядок может быть использован для селекции по логической позиции, например, для селекции первого (*FIRST*), последнего (*LAST*), следующего (*NEXT*), предыдущего (*PREVIOUS*) или *n*-го элемента. Назовем этот тип селекции селекцией по текущей.

Упорядочение может быть фиксированным или произвольным, если порядок не задан явным образом. Например, порядок может быть хронологическим, соответствующим последовательности включения, или определяемым операционными свойствами (например, последовательностью обновления). В этих случаях порядок контролируется СУБД, а не пользователем. Иными словами, не гарантируется, что элемент *x* будет предшествовать элементу *y* или находиться в некоторой фиксированной позиции относительно *y*. С другой стороны, модель данных может допускать определение

порядка в соответствии со значением атрибутов и использование этого порядка для установления текущей.

В языке данных селекция посредством текущей принимает форму операций манипулирования текущими. Мы можем установить текущую в следующую логическую позицию:

SET CURRENCY TO NEXT СТУДЕНТ.

Другим вариантом является манипулирование текущими, подразумеваемое другим действием, например:

GET NEXT СТУДЕНТ.

Здесь действие выборки подразумевает переустановку текущей.

Селекция данных может быть задана значениями, относящимися к атрибутам. Например, можно произвести селекцию всех служащих со значением атрибута *Зарплата*, меньшим, чем 10000. Этот тип селекции называется селекцией по данным.

В языке данных селекция по данным обычно принимает форму условия на значения атрибутов, которое задает критерий селекции. Этот критерий может определять как простые, так и сложные условия отбора данных. Простое условие определяется на одном атрибуте и одном значении атрибута и обычно имеет вид:

<имя атрибута> <операция сравнения> <значение>.

Например, для выборки сразу всех студенток можно воспользоваться следующей операцией:

GET СТУДЕНТ WHERE Пол = 'ж'.

Более сложные критерии селекции могут задаваться выражениями, построенными из простых условий с помощью булевых операторов *И (AND)*, *ИЛИ (OR)*, *НЕ (NOT)*.

Например, для выборки студенток, получающих стипендию, можно воспользоваться следующей операцией:

GET СТУДЕНТ WHERE Пол = 'ж' AND Стипендия \neq 0.

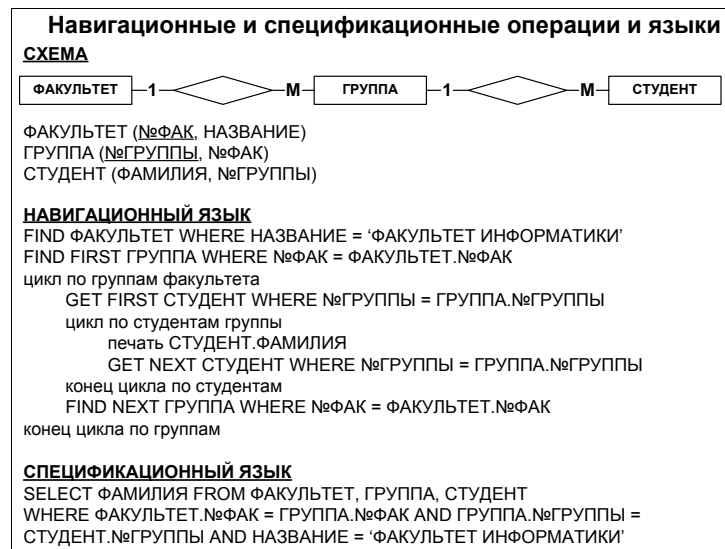
Селекция данных может производиться не только по логической позиции и значению, но и в соответствии с логическими связями между данными. Например, при наличии связи типа *СОБСТВЕННОСТЬ* между типами объектов *СТУДЕНТ* и *АВТОМОБИЛЬ* можно осуществлять селекцию всех студентов, владеющих автомобилями, или всех автомобилей, принадлежащих студентам. Селекция такого рода называется селекцией по связям. Пример на выдачу информации о студентах, владеющих автомобилями:

GET СТУДЕНТ WHERE СТУДЕНТ.Фамилия = АВТОМОБИЛЬ.Фамилия владельца

В зависимости от модели данных и типа языка данных возможны комбинации различных видов селекции. Для примера приведем операцию, которая задает селекцию с использованием всех трех способов:

GET NEXT СТУДЕНТ WHERE Пол = 'ж' AND СТУДЕНТ.Фамилия = АВТОМОБИЛЬ.Фамилия владельца.

В ней осуществляется выборка информации об очередной студентке, владеющей автомобилем.



Определение 2.4.5. Существенный признак, по которому различаются операции и языки данных, определяется характером результата единичной директивы селекции, осуществляемой в ходе выполнения одной операции. Если результат всегда представлен единственным элементом (в зависимости от модели это могут быть значение атрибута, кортеж, сущность, связь и т.д.), полученным при прохождении по логическому пути (т.е. при навигации) в структуре БД, то соответствующие операции называются **навигационными**. Напротив, операции могут специфицировать как бы новую подсхему, определяемую на схеме БД, которой в общем случае соответствует множество элементов, существующих в БД. Такие операции называются **спецификационными**.

Язык, все операции которого являются навигационными, называется **навигационным языком**. Язык, все операции которого являются спецификационными, называется **спецификационным языком**. Однако встречаются языки, которые могут иметь как те, так и другие операции.

На слайде представлены реализации запроса «*Выдать фамилии студентов факультета информатики*» на навигационном и спецификационном языках. В верхней части слайда приведена схема БД, в рамках которой реализуется запрос. В графовой форме представлены три типа объектов (*ФАКУЛЬТЕТ*, *ГРУППА*, *СТУДЕНТ*) и два типа бинарных связей с очевидной семантикой (поэтому их имена мы опустили). Пометки на ребрах говорят о типах этих связей (обе они имеют тип «один ко многим»).

Ниже представлены схемы соответствующих отношений. Как видим, каждому типу объектов соответствует свое отношение, а связи между кортежами-объектами представлены дублированием значений первичных ключей. Понятно, что отношение *ФАКУЛЬТЕТ* понадобится нам, чтобы найти в нем кортеж, относящийся к факультету информатики. Из отношения *СТУДЕНТ* мы получим искомые значения атрибута *Фамилия*. Поскольку напрямую кортежи этих отношений не связаны, нам понадобится отношение *ГРУППА*, имеющее связи с обоими другими отношениями.

Для реализации навигационного запроса нам необходимо по одному разу просканировать кортежи отношений *ФАКУЛЬТЕТ* и *ГРУППА* и перебрать кортежи отношения *СТУДЕНТ* столько раз, сколько групп имеется на факультете информатики. Каждая операция селектирует по одному кортежу, делая его текущим установкой на него соответствующего указателя. Пусть вас не вводит в заблуждение отсутствие типичных для навигационных операций ключевых слов *FIRST*, *LAST*, *NEXT*, *PREVIOUS* у первой операции. Очевидно, что атрибут *Название* отношения *ФАКУЛЬТЕТ* является ключевым, и поэтому кортеж с указанным значением этого атрибута один единственный.

Обратите внимание, что только для кортежей отношения *СТУДЕНТ* осуществляется действие выборки (*GET*), во всех остальных случаях нам достаточно выполнить лишь действие установки текущей (*FIND*).

Понятие текущей есть одно из основных понятий в навигации. В спецификационных операциях текущие на пользовательском уровне не видны. Навигационные операции всегда предполагают селекцию посредством текущей, в то время как для спецификационных операций она не используется.

Таким образом, бросаются в глаза особенности запросов на навигационном языке:

- они всегда представляют собой программу;
- для их формулировки требуется язык программирования с условными конструкциями и конструкциями циклов;
- в них обязательно используется селекция по текущей;
- для них необходимо действие установки текущих.

Ниже приведена одна единственная спецификационная команда выборки, результатом выполнения которой будет таблица с одним столбцом, содержащим искомые значения.

После ключевого слова *SELECT* указан атрибут, значения которого и будут представлены в результирующей таблице. В конструкции *FROM* перечислены все отношения, данные из которых так или иначе участвуют в запросе. После ключевого слова *WHERE* указан критерий селекции кортежей, включающий одну конструкцию селекции по данным и две конструкции селекции по связям.

Как видим, отсутствуют все характерные для навигационных языков черты. Правда, следует отметить, что процедурные языки есть и среди спецификационных. Об этом мы будем говорить, анализируя языки реляционной модели.

Процедуры БД	
Хранятся и выполняются на сервере	
Состоят из разделов:	
<ul style="list-style-type: none"> • условие • действие • результат 	
Классификация процедур:	
<ul style="list-style-type: none"> • Функции агрегирования и другие функции и процедуры общего назначения • Виртуальные атрибуты • Триггеры целостности • Триггеры безопасности • Операторы доступа • Триггеры БД, запускаемые операциями DML (INSERT, UPDATE, DELETE) (прикладные триггеры) • Триггеры БД, запускаемые другими событиями в БД (системные триггеры) 	

Рассмотренные ранее операции следуют достаточно ограниченной схеме, в соответствии с которой вначале выполняется селекция, возможно сопровождаемая установкой текущих, а затем производятся выборка, обновление, включение или удаление. Существует потребность и в других операциях, выполняемых по схемам, отличающимся от упомянутой, или реализующих более глобальные функции.

Возьмем, например, процедурный механизм поддержания целостности, который инициируется операциями изменения данных. Этот механизм может осуществлять проверку большого числа элементов данных, и в силу этого он не укладывается в схему операций манипулирования, которая характеризуется локальностью действий. С другой стороны, здесь нет противоречия с определением операции как действия, переводящего БД из одного состояния в другое.

Обобщенные операции над БД могут быть инкапсулированы в определении процедур базы данных.

Определение 2.4.6. Процедура базы данных представляет собой последовательность операций, выполняемых при определенных условиях. Общность этих условий обеспечивает возможность автоматического, без вмешательства пользователя, инициирования процедур. Действия, выполняемые процедурой, могут быть весьма разнообразны и не ограничиваться частью БД. Процедура посредством информации состояния «уведомляет» систему и пользователя об успешном или неуспешном завершении.

Процедуры БД специфицируются в схеме и в общем случае состоят из трех разделов: условия, действия и уведомления. В разделе условия специфицируется проверка достоверности предпосылок действия. Назначение раздела действий следует из названия. В разделе уведомления указывается, что именно должна передавать процедура при возврате управления и/или задается условие возврата.

Один из видов процедур БД – вычисление значений, которые непосредственно не хранятся в БД, например, вычисление сумм, подсчет числа экземпляров, определение минимума и максимума. Мы будем называть соответствующие процедуры БД **функциями агрегирования**. Во многих моделях функции агрегирования являются конструкциями языка данных.

Второй вид процедур БД – вычисление значения атрибута, например, вычисление возраста человека по заданной текущей дате и дате рождения, что выполняется по запросу о возрасте. Любая процедура БД, вычисляющая значение атрибута, называется **виртуальным атрибутом**. С точки зрения использования виртуальный атрибут обладает

теми же свойствами, что и любой другой атрибут. Однако с ним могут быть связаны различные побочные эффекты и особые ситуации.

Третий вид процедур БД – контроль целостности БД. Верификация ограничений, рассмотренных в предыдущем параграфе, может осуществляться автоматически (декларативный способ определения ОЦ) или с помощью процедур БД, вызываемых явной директивой пользователя или запускаемых при возникновении специфических ситуаций или выполнении специфических операций. Процедуры БД для проверки и/или поддержки ОЦ носят название **триггеров целостности**. Такие процедуры не вырабатывают значений данных, а только «уведомляют» систему об успешном или неуспешном завершении или вызывают принудительное возвращение БД в целостное состояние, согласующееся с определенными ограничениями (что можно трактовать как некоторый побочный эффект).

Четвертый вид процедур БД – обеспечение контроля над доступом. Контроль над доступом не сводится только к парольной защите. Он может быть связан с анализом данных и проверкой полномочий и аутентичности пользователя. Соответствующая процедура БД может выполнять сложные вычисления, по результатам которых могут приниматься важные решения. Например, процедура БД может получать в качестве исходных данных идентификационные параметры пользователя, время, дату, вид действия и вырабатывать сообщения для пользователя, журнала или лица, ответственного за соблюдение правил доступа. Процедуры такого рода называются **триггерами безопасности**.

Процедуры БД могут использоваться также для расширения языка данных операциями, первоначально в нем не предусмотренными. Например, операция сортировки может быть определена как процедура БД, предшествующая выдаче результата спецификационной операции. Другой пример – процедура удаления кортежей-дубликатов, позволяющая модифицировать спецификационную операцию, в результате которой образуется отношение. Такие процедуры называются **операторами доступа**.

Можно определить процедуры БД, которые инициируются операциями модификации одних данных и выполняют косвенные модификации других данных. Если тип связей между отношениями является полным (хотя бы одно из определяемых им отображений полностью определено), то удаление кортежа может привести к запуску процедуры, удаляющей все кортежи, связанные с удаляемым. Такие процедуры называются **триггерами, запускаемыми включением, удалением или обновлением**.

Процедуры БД могут применяться для сбора статистики или для реализации каких-либо других функций администрирования БД. Такие процедуры носят название **процедур администратора БД**. Они не изменяют данные, но могут модифицировать состояние БД.

Подытожим различия между процедурами БД и операциями, рассмотренными в первой части этого параграфа. Во-первых, они не следуют схеме «селекция – действие». Во-вторых, их действия могут захватывать весьма большую область БД. В-третьих, с ними может быть связан ряд побочных эффектов, зависящих от данных. В-четвертых, вызовы этих процедур не выполняются пользователем. В-пятых, они могут реализовать широкий круг действий. И, наконец, процедуры БД обычно описываются в схеме, в то время как операции включаются в пользовательскую программу.

Вопросы и задания к параграфу 2.4

1. Дайте определение «расширенного» состояния БД.
2. Что такое операция над данными?
3. Какие два компонента можно выделить в любой операции над данными?
4. Какие действия над данными предусмотрены в технологии БД?
5. Какими способами можно селектировать данные для выполнения тех или иных действий с ними?

6. Какие два класса операций и языков манипулирования данными выделяются в технологии БД? Каковы их особенности?
7. Что такое процедура БД?
8. Какие виды процедур БД вам известны?