**Дата: 10.12.2020**

**Группа: 19СЗС 1д**

**Наименование дисциплины: Информатика**

**Тема: Основные понятия реляционной БД**

Реляционная база данных состоит из связанных между собой таблиц. Структура таблиц, базы данных в целом и операций над данными, хранящимися в реляционной базе данных, определяются реляционной моделью базы данных. База данных, полностью соответствующая реляционной модели - это абстракция, идеал, к которому нужно стремиться, но на практике не осуществлённый. Реально существующие базы данных, называемые реляционными, имеют некоторые отклонения от реляционной модели. Сложились две терминологии для обозначения очень близких понятий, используемых на практике и в реляционной теории. В табл. 1 приведены соответствия между теоретическими и практическими терминами.

|  |
| --- |
| Таблица 1.Практические и теретические термины |
| **Применяемый напрактике термин** | **Реляционный,теоретический термин** |
| Колонка, столбец, поле | Атрибут. |
| Строка, запись | Кортеж |
| Таблица | Отношение |

Большиство современных СУБД - реляционные, соответственно, создаваемые ими базы данных тоже (почти) реляционные. Отклонения, допускаемые в реляционных СУБД от реляционной модели, объясняются тем, что возникающие на практике трудности при реализации некоторых требований реляционной теории превышают эффект точного следования теории.

В большинстве СУБД разрешается пустая клетка таблицы. Для неё даже придумано обозначение NULL, а каждый атрибут в кортеже должен иметь значение. На практике признак NULL означающий, что в клетку табицы ничего не занесено, широко используется. Разрешается хранить в одной таблице несколько одинаковых строк, а одинаковые множества значений атрибутов в нескольких кортежах не допускаются.

Кортежи в отношении и атрибуты в кортеже не упорядочены, а в таблице строки и столбцы всегда расположены в какой-то последовательности.

### Реляционная модель базы данных

**Модель данных** - это формализованное описание структуры данных и операций над данными.

В 1970 году Кодд предложил применить для построения баз данных реляционную модель. В этой модели структура описывается как набор связанных между собой особого рода таблиц, называемых отношениями (отношение - relation), к которым применяются операции, определённые в реляционной алгебре.

Реляционные базы данных поддаются математическому описанию и, следовательно, сложной автоматической обработке. Самые сложные запросы, формируемые пользователем, реляционная СУБД оптимизирует, используя математический аппарат, и поэтому выполняет поразительно быстро. Благодаря математическому описанию реляционной базы данных, запросы к ней имеют чёткую логическую форму, хорошо стандартизированы (стандарт языка SQL) и, хотя часто очень сложны, но эта сложность минимальна относительно решаемой с помощью запроса задачи.

#### Отношение

Отношение сотоит из заголовка (схемы), кортежей и атрибутов.

**Атрибут** имеет *имя* и *тип* и принимает значения из множества, определяемого типом. Под значением атрибута в зависимости от контекста понимают либо значение на пересечении кортежа и атрибута (клетки таблицы), либо совокупность (вектор-столбец) значений на пересечении атрибута со всеми кортежами отношения. Множество всех возможных значений атрибута называют *доменом*.

В теории баз данных нет единого толкования терминов *тип* и *домен*. Дадим этим терминам определения, которыми в дальнейшем будем пользоваться.

В памяти ЭВМ значения констант и переменных независимо от языка программирования могут храниться в виде

* последовательности символов(строки),
* числа с фиксированной запятой (целого),
* числа с плавающей запятой.

Формы представления данных *строка, целое число* и *число с плавающей запятой* называются *встроенными типами данных*. Одно и то же число может храниться в любой из перечисленных форм, т.е. иметь разный встроенный тип. Представим во всех трёх формах число 15.

Число обычно хранится в строковой форме, когда оно находися в какой-нибыдь фразе, например, "Сегодня 15 марта". Каждая буква или цифра имеет свой код. Единица имеет код *49* в десятичной системе счисления и *00110001* в двоичной. Цифра (символ) 5 имеет двоичный код *00110101*.

В памяти ЭВМ код числа 15 хранится в двоичной системе счисления в виде последовательности из двух байт:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |

 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |

 |
| единица | пять |

Двоичный код числа 15 - *1111*. В однобайтовом формате с фиксированной запятой 15 записывается так:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |

 |
|    ↑знак числа (0 - +; 1 - минус) |

Наиболее сложно число 15 записывается в форме с плавающей запятой. Для этого сначала 15 нужно записать в полулогарифмическом представлении.

[Общая формула](https://i.voenmeh.ru/kafi5/Kam.loc/inform/UprINF_3.html) для полулогарифмического представления числа *X* имеет вид:

 X = q\*sp

где q - мантисса,

 h = sp - характеристика,

 s - основание,

 p - порядок.

Для записи числа в полулогарифмической форме в память ЭВМ принимают

 1) s = 2,

 2)0.5 ≤ |q| < 1.

Число 15 в полулогарифмической форме имеет вид:

 15=(15/16)\*24

В памяти ЭВМ число 15 в форме с плавающей запятой выглядит так:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|

|  |  |  |  |
| --- | --- | --- | --- |
|  0  |  0 | 0 | 0 | 0 | 1 | 0 | 0  |  0  |  1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  |

 |
|

|  |  |  |  |
| --- | --- | --- | --- |
|  + |   порядок p=4               |  + |         мантисса q= 15/16                                                                      |

 |

Над числами с плавающей и фиксированной запятой можно производить арифметические операции, а к числами, для представления которых использован строковый тип, можно применить только операцию конкатенации (слияния). Операция конкатенации во многих языках программирования обозначается значком + как и операция сложения.

**Пример** операций над числами разных типов.

 9/2 = 4 ; 9 + 2 = 11 операции над числами с фиксированной запятой;

 9.0/2.0 = 4.5; 9.0 + 2.0 = 11.0 операции над числами с плавающей запятой;

 "9" + "2" = "92" операция конкатенации.

Нельзя сравнивать числа разных типов. Иногда делают ошибку в базе данных, назначая целочисленный тип первичному ключу и строковый - внешнему ключу. Имея одинаковые значения эти ключи не будут считаться равными друг другу.

Тип данных может трактоваться очень широко. Например, рассмотрим домен, элементы которого - картинки. Тип данных картинки опишем

* форматом (.png, .jpg и т.д.),
* операцией изменения размера картинки,
* темой, на которую подобраны картинки (лес, полевые цветы и т.д.).

Резюмируя всё сказанное, дадим следующее **определение домена и типа**.

**Домен** - это множество элементов, а **тип** - правила по которым сформировано это множество.
В эти правила входят способ представления и отличительные свойства элементов, а также операции над элементами.

**Отношение** для наглядности изобразим в следующем виде:

|  |
| --- |
| https://i.voenmeh.ru/kafi5/Kam.loc/BaseDate/Otnoshenie.png |
| Рис. 1. Отношение |

**Определение отношения**

Будем рассматривать только множества, обладающие следующими свойствами:

1. элементы в множестве не упорядочены;
2. в множестве нет одинаковых элементов.

Назовём заголовком или схемой множество пар названий атрибутов Ai и, может быть, повторяющихся типов Ti (i = 1,2, . . ,n).
Назовём кортежем множество значений атрибутов, по одному значению для каждого атрибута Ai (i = 1,2, . . ,n).
Назовём телом множество из M кортежей.
Отношением называется совокупность заголовка и тела

**Свойства отношения**

1. В отношении отсутствуют одинаковые кортежи.
2. Каждый кортеж содержит ровно по одному значению каждого атрибута.
3. Атрибуты в кортеже не упорядочены.
4. Кортежи в отношении не упорядочены.

Значения атрибутов в кортеже и количество кортежей в отношении могут меняться, поэтому отношение - величина переменная, многомерная. Принято конкретное значение отношения называть просто отношением, а отношение в общем - переменной-отношением.

**Контрольные вопросы:**

1. Структура реляционной БД
2. Что такое модель данных
3. Из чего формируется атрибут
4. В каком формате хранятся значения констант
5. Дайте определение домена и типа

 Преподаватель\_\_\_\_\_\_\_\_\_А.К.Хунарикова