



Лекция 2

Типы данных и домены

Типы данных SQL (1)

- Данные, хранящиеся в столбцах таблиц SQL-ориентированной базы данных, являются типизированными, т. е. представляют собой значения одного из типов данных, определенных в языке SQL или определяемых пользователями путем применения соответствующих средств языка
- Для этого при определении таблицы каждому ее столбцу назначается некоторый тип данных (или домен), и в дальнейшем СУБД должна следить, чтобы в каждом столбце каждой строки каждой таблицы присутствовали только допустимые значения. В этой лекции мы обсудим систему типов языка SQL

Типы данных SQL (2)

- Все допустимые в SQL типы данных, которые можно использовать при определении столбцов, разбиваются на следующие категории:
 - точные числовые типы (exact numerics);
 - приближенные числовые типы (approximate numerics);
 - типы символьных строк (character strings);
 - типы битовых строк (bit strings);
 - типы даты и времени (datetimes);

Типы данных SQL (3)

- типы временных интервалов (intervals);
- булевский тип (Booleans);
- типы коллекций (collection types);
- анонимные строчные типы (anonymous row types);
- типы, определяемые пользователем (user-defined types);
- ссылочные типы (reference types)

Типы данных SQL (4)

- В столбцах таблиц, определенных на любых типах данных, наряду со значениями этих типов, допускается сохранение неопределенного значения, которое обозначается ключевым словом NULL
- В языке определено, что результатом выражений вида $x \text{ а_оп } NULL$, $NULL \text{ а_оп } x$, $NULL \text{ а_оп } NULL$ является NULL для всех арифметических операций а_оп (+, - и т. д.), допустимых для типа данных выражения x
 - выражение $NULL \text{ а_оп } NULL$ является допустимым для любой арифметической операции а_оп

Типы данных SQL (5)

- Также по определению полагается, что значением выражений $x \text{ comp_op } NULL$, $NULL \text{ comp_op } x$, $NULL \text{ comp_op } NULL$ для всех операций сравнения ($=$, $>$, $<$ и т. д.), определенных для типа выражения x , является третье логическое значение unknown
 - выражение $NULL \text{ comp_op } NULL$ является допустимым для любой операции сравнения comp_op

Типы данных SQL (6)

Точные числовые типы (1)

- К категории *точных числовых типов* в SQL относятся те типы, значения которых точно представляют числа
- Типы данных этой категории распадаются на две части: *истинно целые типы* (INTEGER и SMALLINT) и *типы, допускающие наличие дробной части* (NUMERIC и DECIMAL)
- Охарактеризуем эти типы данных более подробно

Типы данных SQL (5)

Точные числовые типы (2) Истинно целые типы

- Тип `INTEGER` служит для представления целых чисел. Точность чисел (число сохраняемых бит) определяется в реализации
 - при определении столбца данного типа достаточно указать просто `INTEGER`.
- Тип `SMALLINT` также служит для представления целых чисел
 - точность определяется в реализации, но она не должна быть больше точности типа `INTEGER`
 - При определении столбца указывается просто `SMALLINT`
- Литералы типов целых чисел представляются в виде строк символов, изображающих десятичные числа; в начале строки могут присутствовать символы «+» или «-» (если символ знака отсутствует, подразумевается «+»)
 - примеры литералов типов `INTEGER` и `SMALLINT`: 1826545, 876

Типы данных SQL (6)

Точные числовые типы (3)

Точные типы, допускающие наличие дробной части (1)

- Тип NUMERIC
- На самом деле, это не просто тип данных, а параметризуемый тип
 - При определении столбца можно указать спецификацию NUMERIC (p, s), где p и s – литералы истинно целого типа, и p задает точность значений (число сохраняемых бит), а s – шкалу (число десятичных цифр в дробной части)
 - Задаваемая шкала не должна быть отрицательной и не должна превышать значение точности

Типы данных SQL (7)

Точные числовые типы (4)

Точные типы, допускающие наличие дробной части (2)

- При определении столбца можно использовать сокращенные формы спецификации типа – NUMERIC и NUMERIC (p)
 - Первая форма предполагает использование точности, определяемое по умолчанию в реализации, и шкалы, равной нулю, а вторая – использование заданной точности и шкалы, равной нулю
 - Допустимые диапазоны значений p и s определяются в реализации

Типы данных SQL (8)

Точные числовые типы (5)

Точные типы, допускающие наличие дробной части (3)

- Тип DECIMAL
- Этот тип аналогичен типу NUMERIC
- Отличие состоит в том, что если при определении столбца типа DECIMAL задается точность p , то на самом деле используется точность m , определяемая в реализации, такая, что $m > p$
 - Шкала всегда устанавливается такой, как явно или неявно (по умолчанию) задается
 - При указании типа столбца можно использовать спецификации DECIMAL, DECIMAL (p) и DECIMAL (p , s)

Типы данных SQL (9)

Точные числовые типы (6)

Точные типы, допускающие наличие дробной части (4)

- Литералы типов точных чисел, допускающих наличие дробной части, представляются в виде строк символов, изображающих десятичные числа, в начале которых могут присутствовать символы «+» или «-» (если символ знака отсутствует, подразумевается «+»), а внутри последовательности цифр может присутствовать символ «.».
- Примеры литералов типов NUMERIC и DECIMAL: 125, 26.36

Типы данных SQL (10)

Приближенные числовые типы (I)

- К категории *приближенных числовых типов* в SQL относятся те типы, значения которых представляют числа приближенным образом
- Приближенные числа представляются в виде пары <мантисса, порядок>, где мантисса состоит из значащих цифр числа, а порядок определяет реальный размер числа
- В реализациях приближенным числовым типам SQL обычно соответствуют типы с плавающей точкой
- В SQL поддерживаются три варианта приближенных числовых типов

Типы данных SQL (II)

Приближенные числовые типы (2)

- Тип REAL
 - Значения типа соответствуют числам с плавающей точкой одинарной точности
 - Точность определяется в реализации, но обычно совпадает с точностью одинарной плавающей арифметики, поддерживаемой на аппаратной платформе, которая используется реализацией
 - При определении столбца указывается просто REAL

Типы данных SQL (12)

Приближенные числовые типы (3)

- Тип DOUBLE PRECISION
 - Точность значений этого типа определяется в реализации, но она должна быть больше точности типа REAL
 - Обычно приближенным числам SQL с двойной точностью соответствуют поддерживаемые аппаратурой числа с плавающей точкой двойной точности
 - При определении столбца указывается просто DOUBLE PRECISION

Типы данных SQL (13)

Приближенные числовые типы (4)

- Тип FLOAT
- Это параметризуемый тип, значение
 - Требуется, чтобы реально обеспечиваемая реализацией точность значений была не меньше p
 - Допустимый диапазон значений параметра p определяется в реализации
 - При определении столбца можно указать либо FLOAT(p), либо просто FLOAT
 - В последнем случае подразумевается точность, определяемая реализацией по умолчанию

Типы данных SQL (14)

Приближенные числовые типы (5)

- Литералы приближенных числовых типов представляются в виде литерала точного числового типа, за которым могут следовать символ «E» и литерал целого числового типа
 - Примеры литералов приближенных числовых типов: 123, 123.12, 123E12, 123.12E12
 - Литеральное выражение xEy представляет значение $x \cdot (10^y)$

Типы данных SQL (15)

Типы символьных строк (1)

- В SQL определены три параметризуемых *типа символьных строк*: CHARACTER (или CHAR), CHARACTER VARYING (или CHAR VARYING, или VARCHAR) и CHARACTER LARGE OBJECT (или CLOB)
- Тип CHARACTER
 - Значениями типа являются символьные строки
 - Конкретный набор допустимых символов определяется в реализации, но, как правило, включает набор символов ASCII
 - При определении столбца допускается использование спецификаций CHARACTER (x) и просто CHARACTER
 - Последний вариант эквивалентен заданию CHARACTER (1)

Типы данных SQL (16)

Типы символьных строк (2)

- После определения столбца типа CHARACTER (x) СУБД будет резервировать место для хранения x символов этого столбца во всех строках соответствующей таблицы
- Если, например, определен столбец типа CHARACTER (8), и в некоторой строке таблицы в него заносится символьная строка длиной пять символов, то реально будут храниться восемь символов, последние три из которых будут пробелами

Типы данных SQL (17)

Типы символьных строк (3)

- Тип CHARACTER VARYING

- При определении столбца допускается использование спецификаций CHARACTER VARYING (x) и просто CHARACTER VARYING
 - Последний вариант эквивалентен заданию CHARACTER VARYING (1)
- Если в некоторой таблице определяется столбец типа CHARACTER VARYING (x), то в каждой строке этой таблицы значения данного столбца будут занимать ровно столько места, сколько требуется для сохранения соответствующей символьной строки (но ни одна такая строка не может состоять более чем из x символов).

Типы данных SQL (18)

Типы символьных строк (4)

- Определен ряд операций, которые можно выполнять над символьными строками
- Перечислим некоторые из них
 - Операция конкатенации (обозначается в виде «||») возвращает символьную строку, произведенную путем соединения строк-операндов в том порядке, в каком они заданы.
 - Функция выделения подстроки (SUBSTRING) принимает три аргумента – строку, номер начальной позиции и длину – и возвращает строку, выделенную из строки-аргумента в соответствии со значениями двух последних параметров.

Типы данных SQL (19)

Типы символьных строк (5)

- Функция UPPER возвращает строку, в которой все строчные буквы строки-аргумента заменяются прописными. Функция LOWER, наоборот, заменяет в заданной строке все прописные буквы строчными.
- Функция определения длины (CHARACTER_LENGTH, OCTET_LENGTH, BIT_LENGTH) возвращает длину заданной символьной строки в символах, октетах или битах (в зависимости от вида вычисляющей функции) в виде целого числа.
- Функция определения позиции (POSITION) определяет первую позицию в строке S, с которой в нее входит заданная строка S1 (если не входит, то возвращается значение нуль)

Типы данных SQL (20)

Типы символьных строк (6)

- Тип CHARACTER LARGE OBJECT
 - Этот тип данных предназначен для определения столбцов, хранящих большие и разные по размеру группы символов
 - При определении столбца задается спецификация CLOB (z), где z задает максимальный размер соответствующей группы символов
 - Максимально возможное значение параметра z определяется в реализации, но, очевидно, что оно должно быть существенно больше максимально возможного значения параметра x, присутствующего в типах CHAR и CHAR VARYING

Типы данных SQL (21)

Типы символьных строк (7)

- Литералы типов символьных строк представляются в виде последовательностей символов, заключенных в одинарные или двойные кавычки
 - В первом случае среди набора символов литерала допускается наличие символов двойной кавычки, а во втором – символов одинарной кавычки
 - Примеры литералов символьных строк: 'ABCDEF', 'Ab"Ctd', "Fbcdef", "ab'cdtF"

Типы данных SQL (23)

Типы битовых строк (2)

- Тип BIT VARYING
- при определении столбца допускается использование только спецификации без умолчания вида BIT VARYING (x), где значение x определяет максимальную длину битовой строки, которую можно хранить в данном столбце
- Над битовыми строками определен ряд операций:
 - битовая конкатенация (обозначается в виде ||) возвращает результирующую битовую строку, полученную путем конкатенации строк-аргументов в том порядке, в котором они заданы

Типы данных SQL (25)

Типы битовых строк (5)

- функция извлечения подстроки из битовой строки
 - синтаксис и семантика этой функции идентичны синтаксису и семантике функции SUBSTRING для символьных строк, за исключением того, что первый аргумент и возвращаемое значение являются битовыми строками.
- функция определения длины (OCTET_LENGTH, BIT_LENGTH) возвращает длину заданной битовой строки в октетах или битах в зависимости от выбранной функции.
- функция определения позиции (POSITION) определяет первую позицию в битовой строке S, с которой в нее входит строка S1
 - если строка S1 не входит в строку S, возвращается значение нуль

Типы данных SQL (32)

Типы даты и времени (6)

- Литералы типа TIME представляются в виде строки TIME 'hh:mm:ss:f...f', где символы h, m, s и f должны изображать десятичные числа
 - например, литерал TIME '16:33-20:333' представляет время суток 16 часов 33 минуты 20 и 333 тысячных секунды

Типы данных SQL (33)

Типы даты и времени (7)

- Тип **TIMESTAMP**
 - значения этого параметризованного типа состоят из **КОМПОНЕНТОВ**
 - значений года, месяца и дня некоторой даты
 - а также **КОМПОНЕНТОВ**
 - значений часа, минуты и секунды некоторого времени суток
 - т. е. каждое значение задает некоторую абсолютную временную метку – отсюда название типа **TIMESTAMP**.
- Число десятичных цифр в значениях-компонентах и ограничения этих значений такие же, как у значений типов **DATE** и **TIME**

Типы данных SQL (34)

Типы даты и времени (8)

- при определении столбца типа **TIMESTAMP** может указываться **TIMESTAMP (p)**
 - значение **p** задает точность долей секунды
- или просто **TIMESTAMP**
 - в этом случае, в отличие от типа данных **TIME**, по умолчанию принимается, что в доли секунды используются шесть десятичных цифр
- максимально допустимое значение **p** определяется в реализации

Типы данных SQL (35)

Типы даты и времени (9)

- Литералы типа `TIMESTAMP` представляются в виде строки `TIMESTAMP 'уууу-мм-дд hh:mm:ss:f...f'`, где символы `у`, `т`, `д`, `h`, `т`, `с` и `f` должны изображать десятичные числа
 - Например, литерал `TIMESTAMP '1949-04-08 16:33-20:333'` представляет временную метку 16 часов 33 минуты 20 и 333 тысячных секунды 8 апреля 1949 г.

Типы данных SQL (36)

Типы даты и времени (9)

- Тип **TIME WITH TIME ZONE**
 - этот тип данных похож на тип **TIME** с тем лишь отличием, что значения типа **TIME WITH TIME ZONE** включают дополнительный компонент — значение, характеризующее смещение соответствующего времени относительно гринвичского времени
 - теперь его называют UTC – universal time coordinated
 - деталей представления этого дополнительного компонента мы касаться не будем.
- Тип **TIMESTAMP WITH TIME ZONE**
 - этот тип данных отличается от типа **TIMESTAMP** тем, что значения типа **TIMESTAMP WITH TIME ZONE** включают дополнительный компонент-значение, характеризующее смещение соответствующего времени относительно гринвичского

Типы данных SQL (37)

Типы временных интервалов (1)

- *Временным интервалом* называется разность между двумя значениями даты или времени
- В SQL определены две категории *типов временных интервалов*: «год-месяц» и «день-время суток»
- Временные интервалы языка SQL не привязываются к начальному и/или конечному значению даты/времени, а описывают только протяженность во времени

Типы данных SQL (38)

Типы временных интервалов (2)

- В общем случае при определении столбца типа временного интервала указывается INTERVAL start (p) [TO end (q)], где в качестве «start» и «end» могут задаваться YEAR, MONTH, DAY, HOUR, MINUTE и SECOND
 - параметр p задает требуемую точность лидирующего поля интервала (число десятичных цифр)
 - параметр q может задаваться только в том случае, когда в качестве end используется SECOND, и указывает точность долей секунды

Типы данных SQL (39)

Типы временных интервалов (3)

- Возможны следующие вариации типов временных интервалов.
- Типы категории «год-месяц»
 - можно определить столбцы следующих типов:
 - INTERVAL YEAR, INTERVAL YEAR (p)
 - значения этих типов – временные интервалы в годах
 - INTERVAL MONTH, INTERVAL MONTH (p)
 - значения этих типов – временные интервалы в месяцах
 - INTERVAL YEAR TO MONTH, INTERVAL YEAR (p) TO MONTH
 - значения этих типов – временные интервалы в годах и месяцах.
 - если значение параметра p не указывается явно, по умолчанию принимается его значение «2»

Типы данных SQL (40)

Типы временных интервалов (4)

- Типы категории «день-время суток»
 - при определении столбца можно использовать следующие комбинации:
 - INTERVAL DAY (p), INTERVAL DAY,
 - INTERVAL DAY (p) TO HOUR, INTERVAL DAY TO HOUR,
 - INTERVAL DAY (p) TO MINUTE, INTERVAL DAY TO MINUTE,
 - INTERVAL DAY (p) TO SECOND (q), INTERVAL DAY TO SECOND (q),
 - INTERVAL DAY (p) TO SECOND, INTERVAL DAY TO SECOND,
 - INTERVAL HOUR (p), INTERVAL HOUR,
 - INTERVAL HOUR (p) TO MINUTE, INTERVAL HOUR TO MINUTE,

Типы данных SQL (41)

Типы временных интервалов (5)

- INTERVAL HOUR (p) TO SECOND (q), INTERVAL HOUR TO SECOND (q),
 - INTERVAL HOUR TO SECOND, INTERVAL MINUTE (p),
 - INTERVAL MINUTE, INTERVAL MINUTE (p) TO SECOND (q),
 - INTERVAL MINUTE TO SECOND (q), INTERVAL MINUTE (p) TO SECOND,
 - INTERVAL MINUTE TO SECOND, INTERVAL SECOND (p, q), INTERVAL
 - SECOND (p), INTERVAL SECOND
- если значение параметра p не указывается явно, по умолчанию принимается его значение «2»
 - значением параметра q по умолчанию является «6»

Типы данных SQL (42)

Типы временных интервалов (6)

- Приведем только один пример литерала одной из разновидностей типа INTERVAL:
 - INTERVAL '10:20' MINUTE TO SECOND – временной интервал в 10 минут и 20 секунд.
- Над значениями темпоральных типов могут выполняться арифметические операции, смысл которых определяется следующей таблицей:

Тип первого операнда	Операция	Тип второго операнда	Тип результата
Datetime	-	Datetime	Interval
Datetime	+ или -	Interval	Datetime
Interval	+	Datetime	Datetime
Interval	+ или -	Interval	Interval
Interval	* или /	Numeric	Interval
Numeric	*	Interval	Interval

Типы данных SQL (43)

Булевский тип (1)

- При определении столбца булевского типа указывается просто спецификация **BOOLEAN**
 - *булевский тип* состоит из трех значений: true, false и unknown
 - соответствующие литералы обозначаются **TRUE**, **FALSE** и **UNKNOWN**
- Поддерживается возможность построения булевских выражений, которые вычисляются в трехзначной логике

Типы данных SQL (44)

Булевский тип (2)

- Таблицы истинности основных логических операций

AND	TRUE	FALSE	UNKNOWN
TRUE	TRUE	FALSE	UNKNOWN
FALSE	FALSE	FALSE	FALSE
UNKNOWN	UNKNOWN	FALSE	UNKNOWN

OR	TRUE	FALSE	UNKNOWN
TRUE	TRUE	TRUE	TRUE
FALSE	TRUE	FALSE	UNKNOWN
UNKNOWN	TRUE	UNKNOWN	UNKNOWN

NOT	
TRUE	FALSE
FALSE	TRUE
UNKNOWN	UNKNOWN

Типы данных SQL (45)

Типы коллекций (1)

- Начиная с SQL:1999, в языке поддерживается возможность использования типов данных, значения которых являются коллекциями значений некоторых других типов
- Обычно под термином *коллекция* понимается одно из следующих образований: массив, список, множество и мультимножество
- В варианте SQL:1999, принятом в 1999 г., были специфицированы только типы массивов
- В стандарте SQL:2003 появилась спецификация типа мультимножества

Типы данных SQL (46)

Типы коллекций (2)

- Любой возможный *тип массива* получается путем применения конструктора типов `ARRAY`
- При определении столбца, значения которого должны принадлежать некоторому типу массива, используется конструкция `dt ARRAY [mc]`, где
 - `dt` специфицирует некоторый допустимый в SQL тип данных, а
 - `mc` является литералом некоторого точного числового типа с нулевой длиной шкалы и определяет максимальное число элементов в значении типа массива
 - в терминологии SQL:1999 это значение называется максимальной кардинальностью массива

Типы данных SQL (47)

Типы коллекций (3)

- В стандарте SQL:1999 многомерные массивы и массивы массивов не поддерживались
- Однако в стандарте SQL:2003 это ограничение было снято, и теперь типом элементов любого типа коллекций может быть любой допустимый в SQL тип данных, кроме самого конструируемого типа коллекции

Типы данных SQL (48)

Типы коллекций (4)

- Элементам каждого значения типа массива соответствуют их порядковые номера, называемые индексами
- Значение индекса всегда должно принадлежать отрезку $[1, m]$
- Значениями типа массива `dt ARRAY [m]` являются все массивы, состоящие из элементов типа `dt`, максимальное значение индекса которых `cs` не превосходит значения `m`
- При сохранении в базе данных значения типа массива занимает столько памяти, сколько требуется для сохранения `cs` элементов
- Обеспечивается доступ к элементам массива по их индексам
 - в частности, можно объявить столбец типа `INTEGER ARRAY [10]` и при вставке строки в соответствующую таблицу задать значение только пятого элемента массива
 - тогда в строку будет занесен массив из пяти элементов, причем первые четыре элемента будут содержать неопределенное значение (`NULL`)

Типы данных SQL (49)

Типы коллекций (5)

- Основными операциями над массивами являются:
 - выборка значения элемента массива по его индексу,
 - изменение некоторого элемента массива или массива целиком и
 - конкатенация (сцепление) двух массивов
 - кроме того, для любого значения типа массива можно узнать значение его cs

Типы данных SQL (50)

Типы коллекций (6)

- При определении столбца таблицы *типа мультимножеств* используется конструкция `dt MULTISSET`, где
 - `dt` задает тип данных элементов конструируемого типа мультимножеств
- Значениями типа мультимножеств являются мультимножества, т. е. неупорядоченные коллекции элементов одного и того же типа, среди которых допускаются дубликаты
 - например, значениями типа `INTEGER MULTISSET` являются мультимножества, элементами которых — целые числа. Примером такого значения может быть мультимножество `{12, 34, 12, 45, -64}`

Типы данных SQL (51)

Типы коллекций (7)

- В отличие от массива, мультимножество является неограниченной коллекцией
 - при конструировании типа мультимножеств не указывается предельная кардинальность значений этого типа
- Однако это не означает, что возможность вставки элементов в мультимножество действительно не ограничена
 - стандарт всего лишь не требует явного объявления границы
- Ситуация аналогична той, которая возникает при работе с таблицами, для которых в SQL не объявляется максимально допустимое число строк

Типы данных SQL (52)

Типы коллекций (8)

- Для типов мультимножеств поддерживаются операции:
 - преобразования типа значения-мультимножества к типу массивов или другому типу мультимножеств с совместимым типом элементов (операция CAST),
 - удаления дубликатов из мультимножества (функция SET),
 - определения числа элементов в заданном мультимножестве (функция CARDINALITY),
 - выборки элемента мультимножества, содержащего в точности один элемент (функция ELEMENT)

Типы данных SQL (53)

Типы коллекций (9)

- Кроме того, для мультимножеств обеспечиваются операции
 - объединения (MULTISET UNION),
 - пересечения (MULTISET INTERSECT) и
 - определения разности (MULTISET EXCEPT)
- Каждая из операций может выполняться в режиме с сохранением дубликатов (режим ALL) или с устранением дубликатов (режим DISTINCT)

Типы данных SQL (54)

Типы коллекций (10)

- Расширенные в SQL:2003 возможности работы с типами коллекций являются принципиально важными
- Даже при наличии определяемых пользователями типов данных и типов массивов SQL:1999 не предоставлял полных возможностей для преодоления исторически присущего реляционной модели данных вообще и SQL в частности ограничения «плоских таблиц»

Типы данных SQL (55)

Типы коллекций (11)

- После появления конструктора типов мультимножеств и устранения ограничений на тип данных элементов коллекции это историческое ограничение полностью ликвидировано
- Мультимножество, типом элементов которого является анонимный строчный тип, представляет собой полный аналог таблицы
- Тем самым, в базе данных допускается произвольная вложенность таблиц
 - возможности выбора структуры базы данных безгранично расширяются

Типы данных SQL (56)

Анонимные строчные типы (12)

- *Анонимный строчный тип* – это конструктор типов ROW, позволяющий производить безымянные типы строк (кортежей)
- Любой возможный строчный тип получается путем использования конструктора ROW
- При определении столбца, значения которого должны принадлежать некоторому строчному типу, используется конструкция ROW (fld1, fld2, ..., fldn),
 - где каждый элемент fld_i, определяющий поле строчного типа, задается в виде тройки fldname, fldtype, fldoptions
 - подэлемент fldname задает имя соответствующего поля строчного типа
 - подэлемент fldtype специфицирует тип данных этого поля.

Типы данных SQL (57)

Анонимные строчные типы (2)

- В качестве типа данных поля строчного типа можно использовать любой допустимый в SQL тип данных, включая типы коллекций, определяемые пользователями типы и другие строчные типы
- Необязательный подэлемент `fldoptions` может задаваться для указания применяемого по умолчанию порядка сортировки, если соответствующий подэлемент `fldtype` указывает на тип символьных строк, а также должен задаваться, если `fldtype` указывает на ссылочный тип
- Степенью строчного типа называется число его полей

Типы данных SQL (58)

Типы, определяемые пользователем (1)

- *Индивидуальные типы (Distinct Types)*
- Можно определить долговременно хранимый, именованный тип данных, опираясь на единственный предопределенный тип
 - например, можно определить индивидуальный тип данных PRICE, опираясь на тип DECIMAL (5, 2)
 - тогда значения типа PRICE представляются точно так же, как значения типа DECIMAL (5, 2)
- Однако в SQL:1999 индивидуальный тип не наследует от своего опорного типа набор операций над значениями

Типы данных SQL (59)

Типы, определяемые пользователем (2)

- Например, чтобы сложить два значения типа PRICE требуется явно сообщить системе, что с этими значениями нужно обращаться как со значениями типа DECIMAL (5, 2)
- Другая возможность состоит в явном определении методов, функций и процедур, связанных с данным индивидуальным типом
 - нет наследования
- Похоже, что в будущих версиях стандарта появятся и другие, более удобные возможности

Типы данных SQL (60)

Типы, определяемые пользователем (3)

- *Структурные типы (Structured Types)*
- Соответствующие возможности SQL:1999 позволяют определять долговременно хранимые, именованные типы данных, включающие один или более атрибутов любого из допустимых в SQL типа данных, в том числе другие
 - структурные типы,
 - типы коллекций,
 - строчные типы и т. д.

Типы данных SQL (61)

Типы, определяемые пользователем (4)

- Стандарт SQL не накладывает ограничений на сложность получаемой в результате структуры данных, однако не запрещает устанавливать такие ограничения в реализации
- Дополнительные механизмы определяемых пользователями методов, функций и процедур позволяют определить поведенческие аспекты структурного типа

Типы данных SQL (62)

Ссылочные типы (1)

- Обеспечивается механизм конструирования типов
 - *ссылочных типов*
- которые могут использоваться в качестве типов столбцов некоторого вида таблиц
 - типизированных таблиц
- Фактически значениями ссылочного типа являются строки соответствующей типизированной таблицы

Типы данных SQL (63)

Ссылочные типы (2)

- Более точно, каждой строке типизированной таблицы приписывается уникальное значение
 - нечто вроде первичного ключа, назначаемого системой или приложением,
- которое может использоваться в методах, определенных для табличного типа, для уникальной идентификации строк соответствующей таблицы
- Эти уникальные значения называются *ссылочными значениями*, а их тип – *ссылочным типом*
- Ссылочный тип может содержать только те значения, которые действительно ссылаются на экземпляры указанного типа
 - т. е. на строки соответствующей типизированной таблицы

Средства определения, изменения определения и отмены определения доменов (1)

- При определении столбцов таблицы требуется явно указывать тип данных каждого столбца
- Для этого можно использовать средства спецификации типа
- Но в SQL поддерживается и другой механизм — механизм доменов
- Домен является долговременно хранимым, именованным объектом схемы базы данных
- Домены можно:
 - создавать (определять),
 - изменять (изменять определения) и
 - ликвидировать (отменять определение)
- Имена доменов можно использовать при определении столбцов таблиц

Средства определения, изменения определения и отмены определения доменов (2)

- Можно считать, что в SQL определение домена представляет собой вынесенное за пределы определения индивидуальной таблицы «родовое» определение столбца, которое можно использовать для определения различных реальных столбцов реальных базовых таблиц
- В языке SQL обеспечиваются средства
 - определения доменов,
 - изменения и
 - отмены существующих определений

Средства определения, изменения определения и отмены определения доменов (3)

- *Определение домена*
- Для определения домена в SQL используется оператор CREATE DOMAIN
- `domain_definition ::= CREATE DOMAIN domain_name [AS] data_type [default_definition] [domain_constraint_definition_list]`
- Здесь `domain_name` задает имя создаваемого домена, `data_type` есть спецификация определяющего типа данных
- В необязательных разделах `default_definition` и `domain_constraint_definition_list` специфицируются значение домена по умолчанию и набор ограничений целостности, которые будут применяться к любому столбцу, определенному на этом домене

Средства определения, изменения определения и отмены определения доменов (4)

- Раздел `default_definition` имеет вид
`DEFAULT { literal | niladic_function | NULL }`
- Здесь
 - `literal` представляет любое допустимое литеральное значение определяющего типа домена,
 - `NULL` обозначает неопределенное значение, а
 - `niladic_function` может задаваться в одной из следующих форм:
 - `USER`
 - `CURRENT_USER`
 - `SESSION_USER`
 - `SYSTEM_USER`
 - `CURRENT_DATE`
 - `CURRENT_TIME`
 - `CURRENT_TIMESTAMP`

Средства определения, изменения определения и отмены определения доменов (5)

- Если в операторе CREATE DOMAIN значение по умолчанию не специфицируется, считается, что такого значения нет
- Однако позже к определению домена можно добавить раздел значения по умолчанию с помощью оператора ALTER DOMAIN
 - Кроме того, этот оператор позволяет удалить раздел значения по умолчанию из существующего определения домена.
- Элемент списка domain_constraint_definition_list имеет вид [CONSTRAINT constraint_name] CHECK (conditional_expression)
- Необязательный раздел CONSTRAINT constraint_name позволяет определить имя нового ограничения целостности
- Если явное указание имени отсутствует, ограничению назначается имя, автоматически генерируемое системой

Средства определения, изменения определения и отмены определения доменов (6)

- Что касается вида условного выражения, служащего собственно ограничением целостности, то в стандарте запрещается лишь прямое или косвенное использование в нем домена, в определение которого входит данное условное выражение.
- Однако наиболее естественным (и наиболее распространенным) видом ограничения домена является следующий:
`CHECK (VALUE IN (list_of_valid_values))`
- Такое ограничение запрещает появление в любом столбце, определенном на данном домене, любого значения определяющего типа, не входящего в список допустимых значений.

Средства определения, изменения определения и отмены определения доменов (7)

- Примеры определений доменов
- В примерах мы будем иметь дело с таблицами служащих (EMP), отделов (DEPT) и проектов (PRO)
- Каждый служащий обладает уникальным номером (EMP_NO) и получает заработную плату (SALARY)
- Определим домены EMP_NO и SALARY
- `CREATE DOMAIN EMP_NO AS INTEGER CHECK (VALUE BETWEEN 1 AND 10000);`
- Номера служащих являются целыми числами, поэтому базовый тип домена EMP_NO есть тип INTEGER
- Кроме того, на значения этого домена устанавливается следующее ограничение: они должны быть больше нуля и не превосходить целое значение 10000

Средства определения, изменения определения и отмены определения доменов (8)

- Домен SALARY определим следующим образом:
- ```
CREATE DOMAIN SALARY AS NUMERIC (10, 2)
DEFAULT 10000.00
CHECK (VALUE BETWEEN 10000.00 AND 20000000.00)
CONSTRAINT SAL_NOT_NULL CHECK (VALUE IS NOT NULL);
```
- Размер заработной платы является значением точного числового типа NUMERIC из десяти десятичных цифр, две из которых составляют дробную часть
- По умолчанию размер заработной платы составляет 10000 руб.
- Установлен диапазон допустимого размера зарплаты от 10000 руб. до 20000000 руб.
- Неопределенное значение зарплаты не допускается
  - на уровне определения домена

## Средства определения, изменения определения и отмены определения доменов (9)

- *Изменение определения домена*
- Для изменения характеристик ранее определенного домена используется оператор SQL ALTER DOMAIN
- Синтаксис этого оператора выглядит следующим образом:
- domain\_alternation ::= ALTER DOMAIN domain\_name domain\_alternation\_action  
domain\_alternation\_action ::= domain\_default\_alternation\_action |  
domain\_constraint\_alternation\_action
- Как видно из синтаксических правил, при изменении определения домена **МОЖНО**
  - выполнить действие по изменению раздела значения по умолчанию либо
  - изменить ограничение домена.

## Средства определения, изменения определения и отмены определения доменов (10)

- Для первого варианта действует следующий синтаксис:
- `domain_default_alteration_action ::= SET default_definition | DROP DEFAULT`
- В случае установки нового значения по умолчанию (SET) это значение автоматически применяется ко всем столбцам, определенным на данном домене
  - более точно, это значение становится новым значением по умолчанию
  - операция не оказывает влияния на состояние существующих строк таблиц базы данных
- В случае отмены раздела значения по умолчанию в определении домена (DROP) существовавшее значение домена по умолчанию становится значением по умолчанию каждого столбца,
  - который определен на данном домене и для которого не специфицировано собственное значение по умолчанию

## Средства определения, изменения определения и отмены определения доменов (11)

- Действие по изменению ограничения домена определяется следующим синтаксисом:
- `domain_constraint_alteration_action ::=`  
`ADD domain_constraint_definition |`  
`DROP CONSTRAINT constraint_name`
- Действие по добавлению нового определения ограничения домена (ADD) приводит к тому, что новое условие добавляется через AND к существующему ограничению домена
- Если к моменту выполнения соответствующего оператора ALTER DOMAIN существуют столбцы некоторых таблиц, текущие значения которых противоречат новому ограничению, то СУБД должна отвергнуть этот оператор ALTER DOMAIN

## Средства определения, изменения определения и отмены определения доменов (12)

- Действие по отмене ограничения домена (DROP) приводит к исчезновению соответствующей части общего ограничения соответствующего домена, что, естественно, не влияет на существующие значения столбцов имеющих таблиц

## Средства определения, изменения определения и отмены определения доменов (13)

- Примеры изменения определения домена
- Немного поупражняемся с доменом SALARY
- Для изменения значения заработной платы по умолчанию с 10000 на 11000 руб. нужно выполнить оператор  
`ALTER DOMAIN SALARY SET DEFAULT 11000.00;`
- Для отмены значения по умолчанию в домене SALARY следует воспользоваться оператором  
`ALTER DOMAIN SALARY DROP DEFAULT;`
- Если к определению домена SALARY требуется добавить ограничение
  - например, запретить значение зарплаты, равное 15000 руб.
- необходимо выполнить оператор  
`ALTER DOMAIN SALARY ADD CHECK (VALUE <> 15000.00);`

## Средства определения, изменения определения и отмены определения доменов (14)

- Наконец, если требуется отменить (именованное!) ограничение целостности, препятствующее наличию неопределенных значений в столбцах, которые определены на домене SALARY, то нужно выполнить оператор  
`ALTER DOMAIN SALARY DROP CONSTRAINT SAL_NOT_NULL;`



## Средства определения, изменения определения и отмены определения доменов (15)

- *Отмена определения домена*
- Чтобы отменить ранее созданное определение домена, нужно воспользоваться оператором DROP DOMAIN в следующем синтаксисе:  
DROP DOMAIN domain\_name {RESTRICT | CASCADES}
- Если в операторе указано RESTRICT, и если соответствующий домен использован в определении некоторого столбца, в определении некоторого представления или в определении ограничения целостности, то оператор DROP DOMAIN отвергается
- В противном случае определение домена ликвидируется.

## Средства определения, изменения определения и отмены определения доменов (16)

- Если в операторе DROP DOMAIN указано CASCADES, то оператор выполняется всегда
- При этом уничтожаются все представления и ограничения целостности, в определении которых использовалось имя данного домена
- Столбцы, определенные на этом домене, автоматически переопределяются следующим образом:
  - считается, что каждый такой столбец теперь относится к определяющему типу уничтожаемого домена;
  - если у столбца не было определено собственное значение по умолчанию, то считается, что теперь у него имеется такое значение по умолчанию, совпадающее со значением по умолчанию уничтожаемого домена;
  - каждый столбец наследует все ограничения уничтожаемого домена

## Неявные и явные преобразования типа или домена (1)

- В SQL поддерживается совместимость некоторых типов данных за счет *неявного преобразования* значений одного типа к значениям другого типа данных
- например, при необходимости FLOAT неявно приводится к DOUBLE
- Опишем наиболее важные правила совместимости типов, принятые в SQL:1999
- Начнем с определения приводимости типов
- Тип данных A приводим к типу данных B в том и только в том случае, когда в любом месте, где ожидается значение типа B, может быть использовано значение типа A

## Неявные и явные преобразования типа или домена (2)

- Основные правила приводимости типов состоят в следующем.
- *Типы символьных строк*
  - Тип CHARACTER (x) приводим к любому типу CHARACTER (y), если  $y \geq x$ . Типы VARCHAR (x) и CHARACTER (x) приводимы к любому типу VARCHAR (y), если  $y \geq x$ . Типы CHARACTER (x) и VARCHAR (x) приводимы к любому типу CLOB.
- *Типы битовых строк*
  - Тип BIT (x) приводим к любому типу BIT (y), если  $y \geq x$ . Типы BIT VARYING (x) и BIT (x) приводимы к любому типу BIT VARYING (y), если  $y \geq x$ .

## Неявные и явные преобразования типа или домена (3)

- *Типы BLOB*
  - Тип BLOB ( $x$ ) приводим к любому типу BLOB ( $y$ ), если  $y \geq x$ .
- *Типы точных чисел*
  - Тип EN ( $p_1, s_1$ ) приводим к любому типу EN ( $p_2, s_2$ ), у которого  $s_2 \geq s_1$  и  $p_2$  определяется в реализации.
  - Тип EN ( $p, s$ ) приводим к любому типу приближенных чисел AN ( $p_1$ ), где  $p_1$  определяется в реализации.
- *Типы приближенных чисел.*
  - Тип AN ( $p_1$ ) приводим к любому типу AN ( $p_2$ ), если  $p_2 \geq p_1$

## Неявные и явные преобразования типа или домена (4)

- Неявные преобразования типов не всегда удобны, недостаточно гибки и иногда могут вызывать ошибки
- Поэтому число допустимых неявных преобразований типов в SQL весьма ограничено
- Однако в SQL существует специальный оператор CAST, с помощью которого можно *явно* преобразовывать типы или домены в более широких пределах допускаемых преобразований
- Конструкция имеет следующий синтаксис:  
CAST ({scalar-expression | NULL } AS {data\_type | domain\_name})

## Неявные и явные преобразования типа или домена (5)

- Оператор преобразует значение заданного скалярного выражения к указанному типу или к базовому типу указанного домена
- Результатом применения оператора CAST к неопределенному значению является неопределенное значение
- Для значений, отличных от неопределенных, в стандарте приводятся подробные правила выполнения преобразований, которые интуитивно понятны

## Неявные и явные преобразования типа или домена (5)

- Поясним действие оператора CAST в наиболее важных случаях
- Обозначим:
  - EN – точные числовые типы (Exact Numeric)
  - AN – приближенные числовые типы (Approximate Numeric)
  - C – типы символьных строк (Character)
  - FC – типы символьных строк постоянной длины (Fixed-length Character)
  - VC – типы символьных строк переменной длины (Variable-length Character)
  - B – типы битовых строк (Bit String)



## Неявные и явные преобразования типа или домена (6)

- FB – типы битовых строк постоянной длины (Fixed-length Bit String)
- VB – типы битовых строк переменной длины (Variable-length Bit String)
- D – тип Date
- T – типы Time
- TS – типы Timestamp
- YM – типы Interval Year-Month
- DT – типы Interval Day-Time

## Неявные и явные преобразования типа или домена (6)

- Пусть TD – это тип данных, к которому производится преобразование, а SD – тип данных операнда
- Тогда допустимы следующие комбинации
  - «да» означает безусловную допустимость, «нет» – безусловную недопустимость и «?» – допустимость с оговорками

| SD | TD  |     |    |    |     |     |     |     |     |     |     |
|----|-----|-----|----|----|-----|-----|-----|-----|-----|-----|-----|
|    | EN  | AN  | VC | FC | VB  | FB  | D   | T   | TS  | YM  | DT  |
| EN | Да  | Да  | Да | Да | Нет | Нет | Нет | Нет | Нет | ?   | ?   |
| AN | Да  | Да  | Да | Да | Нет | Нет | Нет | Нет | Нет | Нет | Нет |
| C  | Да  | Да  | ?  | ?  | Да  | Да  | Да  | Да  | Да  | Да  | Да  |
| B  | Нет | Нет | Да | Да | Да  | Да  | Нет | Нет | Нет | Нет | Нет |
| D  | Нет | Нет | Да | Да | Нет | Нет | Да  | Нет | Да  | Нет | Нет |
| T  | Нет | Нет | Да | Да | Нет | Нет | Нет | Да  | Да  | Нет | Нет |
| TS | Нет | Нет | Да | Да | Нет | Нет | Да  | Да  | Да  | Нет | Нет |
| YM | ?   | Нет | Да | Да | Нет | Нет | Нет | Нет | Нет | Да  | Нет |
| DT | ?   | Нет | Да | Да | Нет | Нет | Нет | Нет | Нет | Нет | Да  |

## Неявные и явные преобразования типа или домена (7)

- По поводу ячеек таблицы, содержащих знак вопроса, необходимо сделать несколько оговорок:
  - если TD – интервал и SD – тип точных чисел, то TD должен содержать единственное поле даты-времени;
  - если TD – тип точных чисел и SD – интервал, то SD должен содержать единственное поле даты-времени;
  - если SD – тип символьных строк и TD – тип символьных строк постоянной или переменной длины, то набор символов SD и TD должен быть одним и тем же