

## Предложения SQL

- определения данных (DDL)

ALTER INDEX	DATABASE
ALTER TABLE	DROP DATABASE
CLOSE DATABASE	DROP INDEX
CREATE DATABASE	DROP SYNONYM
CREATE INDEX	DROP TABLE
CREATE SCHEMA	DROP VIEW
CREATE SYNONYM	RENAME COLUMN
CREATE TABLE	RENAME TABLE
CREATE VIEW	

- манипулирования данными (DML)

INSERT	SELECT
DELETE	UPDATE

- манипулирования данными (DML)

INSERT

SELECT

DELETE

UPDATE

- работа с курсорами

CLOSE

FLUSH

DECLARE

OPEN

FETCH

SQL включает 47 типов предложений, которые могут быть условно разделены на следующие группы:

- определения данных - используются для создания базы данных и определения ее структуры;
- манипулирования данными - для выборки, добавления, модификации и удаления данных из базы данных;
- работы с курсорами - определения, открытия и т.п. действия с курсорами.

- динамическое управление

DESCRIBE

EXECUTE

EXECUTE IMMEDIATE

FREE

PREPARE

- доступа к данным (DCL)

GRANT

LOCK TABLE

REVOKE

SET ISOLATION

SET LOCK MODE

UNLOCK TABLE

- целостности данных

BEGIN WORK

COMMIT WORK

CREATE AUDIT

DROP AUDIT

ROLLBACK WORK

ROLLFORWARD DATABASE

SET LOG

START DATABASE

- оптимизации очередей
- сохраненные процедуры
- дополнительные предложения

- динамического управления - для управления ресурсами в процессе функционирования системы;
- доступа к данным - определение прав и режимов доступа к данным;
- целостности данных - обеспечение целостности базы данных;
- оптимизации очередей - используются для получения информации требуемой для уменьшения времени обращения к данным;
- дополнительные предложения которые не входят в описанные ранее группы, но являются предложениями SQL и поддерживаются некоторыми продуктами INFORMIX (такие как WHENEVER, INFO, OUTPUT).

## Синтаксические соглашения SQL

```
SELECT ENAME, SAL*12, MONTHS_BETWEEN(HIREDATE, SYSDATE) FROM  
EMP
```

```
SELECT ENAME,  
SAL*12,  
MONTHS_BETWEEN(HIREDATE, SYSDATE)  
FROM EMP
```

Оператор SQL может включать один или более знаков табуляции, возвратов каретки, пробелов и комментариев в любом месте оператора, где допустим пробел. Поэтому, показанные операторы будут интерпретироваться одинаково.

Регистр не имеет значения в зарезервированных словах, идентификаторах и параметрах.

Оператор SQL может включать один или более знаков табуляции, возвратов каретки, пробелов и комментариев в любом месте оператора, где допустим пробел. Поэтому, показанные операторы будут интерпретироваться одинаково.

Регистр не имеет значения в зарезервированных словах, идентификаторах и параметрах. Однако в символьных литералах регистр имеет значение.

## Идентификатор



- |               |   |  |
|---------------|---|--|
| цифра         | - | целое в интервале 0 - 9                    |
| буква         | - | символ верхнего/нижнего регистра от а до z |
| подчеркивание | - | символ подчеркивания (_)                   |

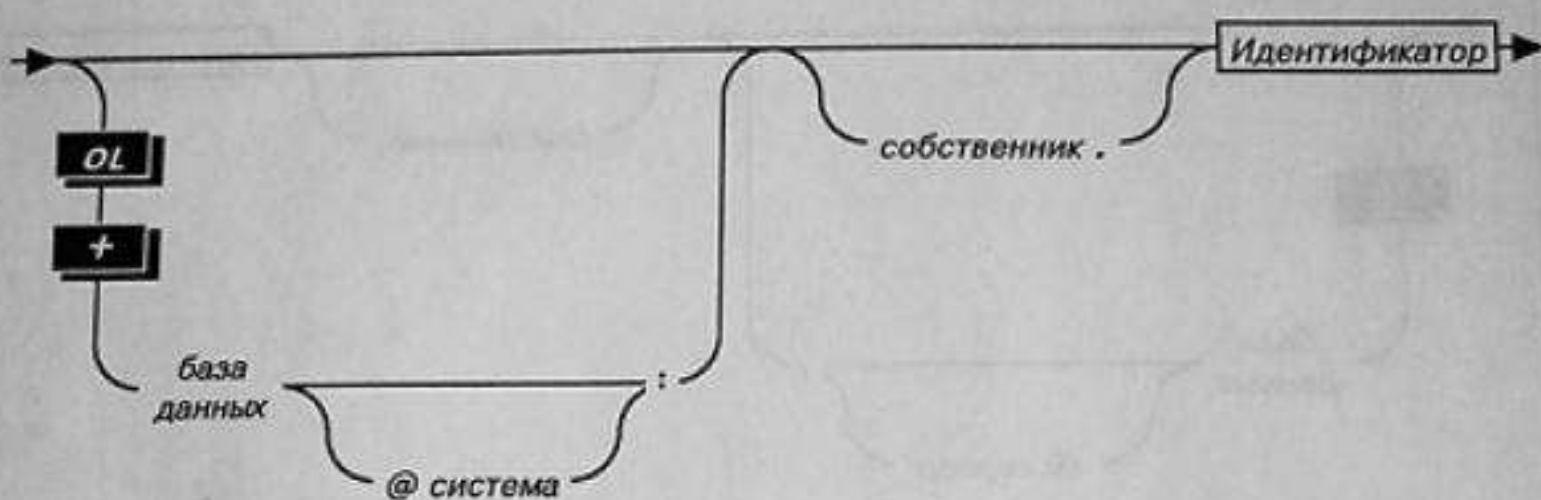


Идентификаторы используются для спецификации имен объектов базы данных:

- имена ограничений
- имя базы данных
- имя индекса
- имя синонима
- имя таблицы и т.д.

Идентификатор может содержать до 18 символов. В качестве идентификатора нельзя использовать зарезервированные слова.

## Имя таблицы



- база данных** - имя базы данных, где размещается таблица
- система** - имя системы INFORMIX-OnLine, где размещается база данных. Символ @ обязательно предшествует имени системы.
- собственник** - имя пользователя-собственника таблицы. В ANSI базах данных необходимо использовать явное указание имени собственника при ссылке на чужие таблицы.

Имя таблицы используется во всех предложениях SQL при ссылках на таблицы.

Пример спецификации таблицы:

... в SQL предложениях SQL при ссылках на таблицы.  
Пример спецификации таблицы:

`empinfo@personnel:emp_names`

Комбинация имен собственника и таблицы должна быть уникальной в базе данных.

**ANSI** Система преобразует имя собственника к верхнему регистру. Для того чтобы исключить преобразование, задайте имя собственника в символах ":

`SELECT * FROM cath1.customer`

`SELECT * FROM "nancy".customer`

BYTE	- хранение любых двоичных данных
CHAR	- хранение любой строки букв, цифр, символов
CHARACTER	- синоним CHAR
DATE	- хранение календарных дат
DATETIME	- хранение календарных дат в сочетании со значением времени
DECIMAL	- хранение чисел с определяемой размерностью и точностью
DOUBLE PRECISION	- синоним для FLOAT
INT	- синоним для INTEGER
INTEGER	- хранение целых чисел
INTERVAL	- хранение интервалов времени
MONEY	- хранение денежных величин
NUMERIC	- синоним для DECIMAL
REAL	- синоним для SMALLFLOAT
SERIAL	- хранение последовательных целых чисел
SMALLFLOAT	- хранение чисел с плавающей точкой одинарной точности
SMALLINT	- хранение целых чисел в диапазоне от -32.767 до 32.767
TEXT	- хранение любых типов текстовых данных
VARCHAR	- хранение символьных данных переменной длины

**CHAR [ (n) ] CHARACTER [ (n) ]**

Используется для хранения строк символов

Максимальная длина поля 32767

**n** - размер поля

При чтении и записи значения поля типа CHAR(n) всегда передается n байтов. Если значение короче, чем n то строка заполняется в конце пробелами. Если длина строки больше n, то она усекается.

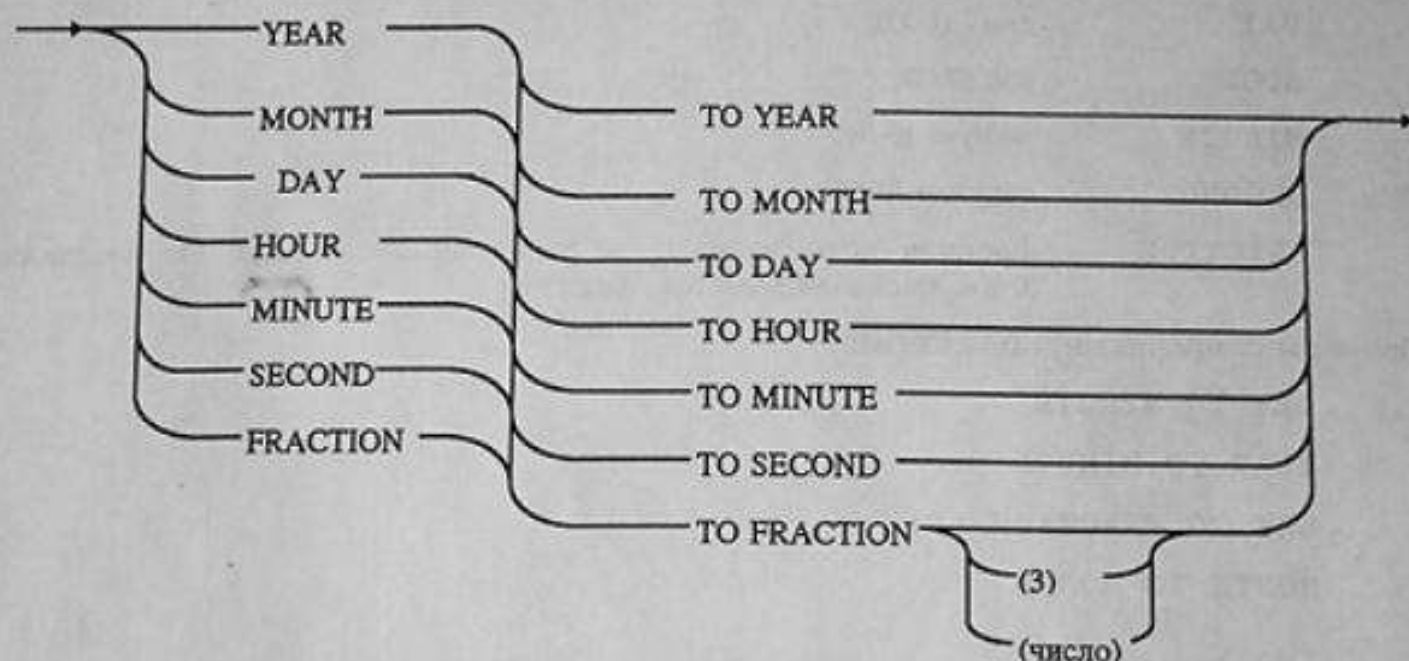
## Тип данных DATE

- Используется для хранения календарных дат
- Календарная дата хранится как целое число количества дней с 31 декабря 1899 г.
- По умолчанию формат даты: mm/dd/yyyy, где
  - mm - месяц (1-12)
  - dd - день (1-31)
  - yyyy - год (0001-9999)

Так как значение календарной даты хранится в виде целого числа, его можно использовать в арифметических выражениях. К примеру, можно вычесть дату из даты. Результат, положительная или отрицательная целая величина, определяет интервал в днях между двумя датами.

Тип данных DATE требует 4 байта для хранения.

## Тип данных DATETIME



### Синтаксис:

DATETIME *наибольший\_квалификатор* TO *наименьший\_квалификатор*

Спецификация DATETIME определяет наибольшую и наименьшую единицу времени в столбце или значении типа DATETIME. Тип данных DATETIME хранит выражение времени как календарную дату и время для. При необходимости...



DATETIME представляется в виде непрерывной последовательности полей даты, которые соответствуют единицам времени, подлежащих хранению.

Столбец типа DATETIME не обязательно должен включать все позиции с YEAR до FRACTION, это может быть подмножество или вообще единичное поле.

Спецификация DATETIME может быть использована в :

- описании типа данных
- выражениях





Ключевые слова для задания полей даты:

YEAR	-	год(1-9999)
MONTH	-	месяц (1-12)
DAY	-	день (1-31)
HOURL	-	час (0-23)
MINUTE	-	минута (0-59)
SECOND	-	секунда (0-59)
FRACTION	-	фракция (доля секунды)- до 5 десятичных позиций. По умолчанию - 3, т.е. тысячная доля секунды.

Примеры спецификаций DATETIME:

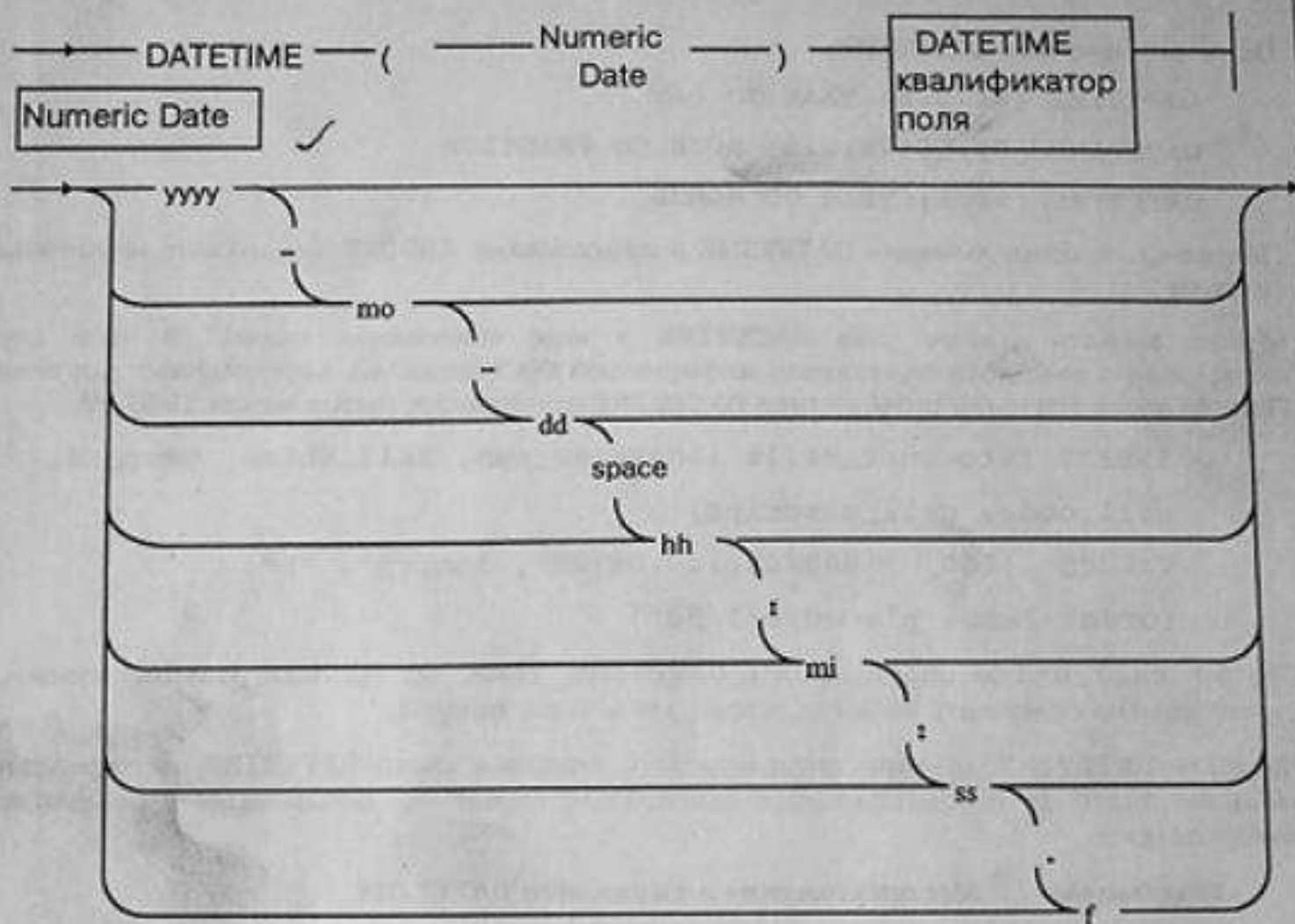
DAY TO MINUTE

YEAR TO MINUTE

DAY TO FRACTION (4)

MONTH TO MONTH

## Литерал DATETIME



Литерал DATETIME используется как значение DATETIME и применяется в :

- предложениях INSERT
- предложениях SELECT
- условиях
- выражениях

yyyy - год - четыре цифры

mo - месяц - две цифры

dd - день - две цифры

space - разделитель - пробел

hh - часы - две цифры

mi - минуты - две цифры

ss - секунды - две цифры

f - доли секунды - до пяти цифр

Примеры литералов DATETIME:

```
DATETIME (89-3-6) YEAR TO DAY
```

```
DATETIME (09:55:30.825) HOUR TO FRACTION
```

```
DATETIME (92-5) YEAR TO MONTH
```

При использовании литерала DATETIME в предложении INSERT, он должен заключаться в символы ".

Можно вводить данные типа DATETIME в виде символьной строки. В этом случае, символьная строка должна включать информацию для всех полей, определенных для столбца. Пример ввода значения столбца типа DATETIME при помощи предложения INSERT:

```
INSERT into cust_calls (customer_num, call_dtime, user_id,  
call_code, call_descript)  
VALUES (100, "1990-08-14 08:45", "maryj", "D",  
"Order late- placed 6/1/90")
```

Столбец call\_dtime определен как DATETIME YEAR TO MINUTE. Поэтому символьная строка должна содержать поля год, месяц, день, часы, минуты.

Можно вводить данные типа DATETIME в виде символьной строки. В этом случае символьная строка должна включать информацию для всех полей, определенных для столбца. Пример ввода значения столбца типа DATETIME при помощи предложения INSERT:

```
INSERT into cust_calls (customer_num, call_dtime, user_id,  
call_code, call_descript)  
VALUES (100, "1990-08-14 08:45", "maryj", "D",  
"Order late- placed 6/1/90")
```

Столбец call\_dtime определен как DATETIME YEAR TO MINUTE. Поэтому символьная строка должна содержать поля год, месяц, день, часы, минуты.

Литералу DATETIME должно предшествовать ключевое слово DATETIME, непосредственно вводимое значение и спецификации полей. Поля имеют вид целых чисел с разделителями между полями.

Разделитель	Местоположение в выражении DATETIME
тире	между годом, месяцем и днем
пробел	между днем и часом
двоеточие	между часом и минутами, минутами и секундами
десятичная	точка между секундами и долями секунды

Пример:

90 - 08 - 16 12 : 42 : 06 . 011  
↑ ↑ ↑ ↑  
год месяц день часы минуты секунды доли секунды



Если вводится значение с не всеми полями, которые были определены для столбца, то недостающие поля даты автоматически добавляются в соответствии со следующими правилами:

- если опущено более значимое поле, или же это поле имеет большую величину, чем описано, то оно автоматически заполняется текущей датой.
- если опущены менее значимые поля, то они заполняются нулями (или 1 для MONTH и DAY).
- если же символьная строка содержит не все поля, то возвращается код ошибки.

Все поля в DATETIME - двухсимвольные, только доли секунды - до 5 символов и год - четыре цифры. Можно использовать следующую формулу для расчета количества байтов необходимых для хранения величины типа DATETIME:

$$(\text{общее количество цифр для всех полей}) / 2 + 1$$

Например, для представления величины типа YEAR TO DAY требуется 8 цифр (4 - для года, 2 - для месяца, 2 - для дня). Эти данные требуют для хранения 5 байт:  $8 / 2 + 1 = 5$

## Тип данных DEC, DECIMAL [(p[,s])], NUMERIC

- Используется для хранения десятичных чисел в форматах с плавающей и фиксированной точкой
- Максимальная длина поля 32 значащих цифры

p - общее число значащих цифр

s - число цифр дробной части

Ключевые слова DEC, DECIMAL, NUMERIC - синонимы.

Если определены обе величины p и s, то десятичное число имеет представление с фиксированной точкой. Все числа по абсолютной величине меньше чем  $0.5 \cdot 10^{-s}$  представляются как ноль.

Наибольшая величина, которую можно хранить без ошибки:  $10^{p-s} - 10^{-s}$ .

Если не указано значение p - по умолчанию принимается DECIMAL(16), десятичное число с плавающей точкой с 16 позициями. Если не задано s - десятичное число представляется в формате с плавающей точкой и абсолютным значением в диапазоне от  $10^{-128}$  до  $10^{126}$ .

Для оценки числа байтов, необходимых для хранения типа DECIMAL можно использовать следующую формулу:



- Максимальная длина поля 32 значащих цифры

$p$  - общее число значащих цифр

$s$  - число цифр дробной части

Ключевые слова DEC, DECIMAL, NUMERIC - синонимы.

Если определены обе величины  $p$  и  $s$ , то десятичное число имеет представление с фиксированной точкой. Все числа по абсолютной величине меньше чем  $0.5 \cdot 10^{-s}$  представляются как ноль.

Наибольшая величина, которую можно хранить без ошибки:  $10^{p-s} - 10^{-s}$ .

Если не указано значение  $p$  - по умолчанию принимается DECIMAL(16), десятичное число с плавающей точкой с 16 позициями. Если не задано  $s$  - десятичное число представляется в формате с плавающей точкой и абсолютным значением в диапазоне от  $10^{-128}$  до  $10^{126}$ .

Для оценки числа байтов, необходимых для хранения типа DECIMAL можно использовать следующую формулу:

$$p/2+1,$$

полученное значение округляется до целого числа.

Например, для DECIMAL(16,2) требуется 9 байт для хранения ( $16/2+1=9$ ).

## Тип данных DOUBLE PRECISION, FLOAT[ (n) ]

- Используется для хранения чисел в формате с плавающей точкой
- Максимальная длина поля 16 значащих цифр

n - число значащих цифр в представлении числа

DOUBLE PRECISION, FLOAT - синонимы.

Диапазон значений соответствует диапазону чисел двойной точности в языке программирования С на Вашем компьютере. n может быть в пределах от 1 до 16.

Для хранения данных типа FLOAT требуется 8 байтов.

## Тип данных INTERVAL

Класс интервалов YEAR - MONTH:

YEAR

- количество лет

MONTH

- количество месяцев

Класс интервалов DAY-TIME :

DAY

- количество дней

HOUR

- количество часов

MINUTE

- количество минут

SECOND

- количество секунд

FRACTION

- десятые доли секунд до  
5 цифр

Поля типа данных INTERVAL хранят величины интервалов времени. Тип данных INTERVAL делится на два класса:

- (year to month interval)
- (day time interval).

которые соответствуют величинам года-месяцы и дни, часы, минуты, секунды и доли секунды.

Величина типа INTERVAL содержит либо одно значение, либо непрерывную последовательность значений компонент времени. Тип данных INTERVAL имеет следующий синтаксис:

INTERVAL *наибольший\_квалификатор(n)* TO *наименьший\_квалификатор(n)*

где *квалификаторы* принадлежат одному классу (см.слайд).

Не допускается использовать одновременно *квалификаторы* полей из разных классов.



Значение, вводимое в столбец типа INTERVAL необязательно должно содержать все определенные поля времени. Например, можно ввести значение с полями времени часы-секунды в столбец, определенный как день-секунды. Однако, значение должно содержать непрерывную последовательность полей времени.

При определении типа данных INTERVAL, необходимо задать два поля (минимальное и максимальное), также как для типа данных DATETIME. Дополнительно можно указывать точность для первого поля (и последнего, если это доли секунды). Если оба поля типа доли\_секунды, то можно определить точность последнего поля.

Старший квалификатор может иметь точность до 9 цифр (если это доли секунды - то только 5). Например, для определения интервалов DAY TO HOUR, содержащих более 99 дней необходимо задать: INTERVAL DAY(3) TO HOUR.

Значения типа INTERVAL имеют те же разделители, что и значения типа DATETIME (см. ранее).

Значение типа INTERVAL можно также ввести в виде символьной строки. Символьная строка должна содержать информацию для всех полей столбца.

Пример ввода значения в виде символьной строки:

```
INSERT INTO manufact(manu_code, manu_name, lead_time)
VALUES ("BRO", "Dall-Rackquet", "160")
```

В примере, столбец lead-time определен как тип INTERVAL DAY(3) TO DAY, поэтому только число дней требуется для ввода в столбец. Необходимо отметить, что если символьная строка содержит не все требуемые поля для столбца типа INTERVAL, то возникает ошибка.

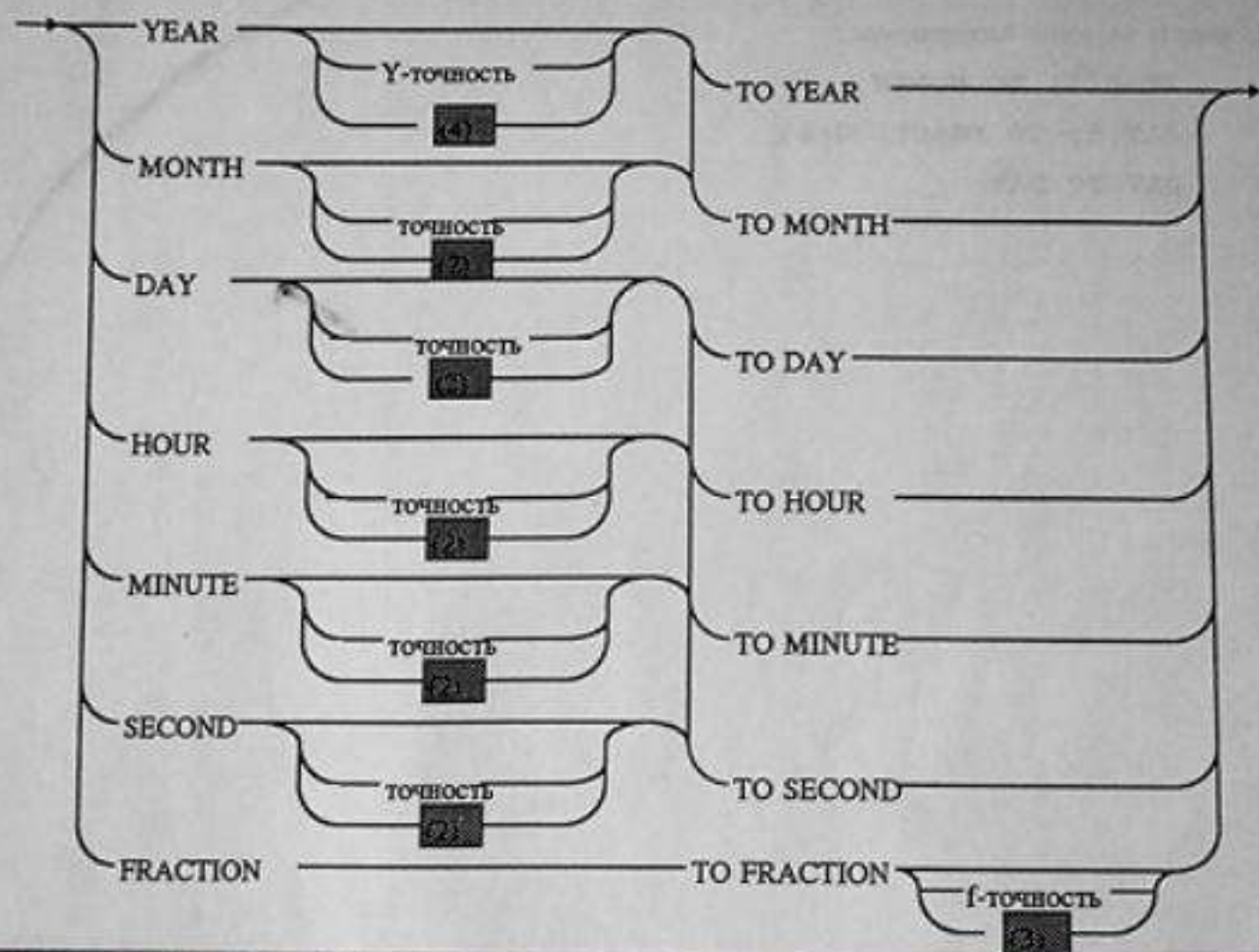
По умолчанию все поля в INTERVAL являются двухсимвольными за исключением полей год и доли секунды. Поле года хранится как четыре пифры, а доли секунды - до 5 пифр. Можно

По умолчанию все поля в INTERVAL являются двухсимвольными за исключением полей год и доли секунды. Поле года хранится как четыре цифры, а доли секунды - до 5 цифр. Можно использовать следующую формулу (с округлением до целого числа байт) для вычисления количества байт, необходимых для хранения величины типа INTERVAL:

$$(\text{общее количество цифр для всех полей}) / 2 + 1$$

Например, для столбца типа YEAR <sup>4</sup>TO <sup>5</sup>MONTH требуется 6 цифр, т.е. 4 байта памяти.

## Определитель поля INTERVAL



- f-точность* - точность - максимальное число цифр, которые можно использовать в поле долей секунды. По умолчанию - 3, максимально - 5.
- точность* - число цифр наибольшего числа месяцев, дней, часов или минут, которое интервал может содержать. По умолчанию - 2, максимум - 9.
- Y-точность* - число цифр наибольшего числа лет которое может содержать интервал. По умолчанию - 4, максимум - 9.

Следующие примеры являются определением типа данных INTERVAL, оба YEAR TO MONTH. В первом примере интервал может быть до 999 лет, так как определена точность в 3 цифры. Во втором примере интервал до 9999 лет - использована точность по умолчанию.

YEAR(3) TO MONTH



## Тип данных MONEY [ (p, [s]) ]

- Используется для хранения денежных единиц
- Максимальная длина поля 32 значащих цифры

p - общее число значащих цифр

s - число цифр дробной части

В отличие от типа DECIMAL, тип MONEY всегда обрабатывается как десятичное число с фиксированной точкой. Определение MONEY(p) рассматривается как DECIMAL(p, 2). Если p и s не указаны, то по умолчанию MONEY воспринимается как DECIMAL(16, 2).

Значения столбцов типа MONEY отображаются с символом соответствующей валюты (денежного знака), по умолчанию доллара, и символом десятичной точки. Можно использовать следующую формулу для вычисления количества байт для хранения величины типа MONEY (округляемую до целого числа байт):  $p/2+1$

Например, для хранения данных столбца типа MONEY(16, 2) требуется 9 байт  $(16/2+1)$ .

## Тип данных SERIAL [ (n) ]

- Используется для задания последовательности целых чисел

n - начальное значение последовательности

Тип данных SERIAL используется для задания последовательности целых чисел, определяемых системой автоматически при добавлении строки в таблицу. В таблице может быть определен только один столбец типа SERIAL.

По умолчанию, n равно 1, но n можно задать при создании/модификации таблицы. Наибольшее значение столбца типа SERIAL - 2147483647.

Однажды определенный, столбец типа SERIAL не может быть изменен. Тем не менее, можно добавить значение в столбец типа SERIAL (в предложении INSERT) или изменить начальный номер (в предложении ALTER TABLE) в том случае, если это значение не дублирует существующие в таблице. При добавлении строки и/или определении начального значения столбца типа SERIAL, система автоматически определяет следующий номер по формуле:

$(\text{максимальное значение в столбце SERIAL}) + 1$

Например, в таблицу customer добавляется строка, где столбец customer\_num типа SERIAL имеет значение 50 и наибольшее значение в столбце равно 128. В этом случае следующий customer\_num будет равен 129.

## **OL** Тип данных VARCHAR [ (m[, r]) ]

cust_num serial	cust_name char(35)	address varchar(100)
1218	Joe Alabama	.126 Sysamore st.
1219	Hannar Rega	1811 West Broadway Suite 37
1220	Martin Tall	35 W.11th st.

- Используется для хранения символьных строк переменной длины
- Максимальный размер поля 255 символов (байтов)

m - максимальный размер поля

r - минимальный размер резервируемого пространства

Для типа `VARCHAR` должен быть определен максимальный размер ( $m$ ) столбца.  $1 \leq m \leq 255$ .  
Спецификация размера резервируемого пространства ( $r$ ) не обязательна.  $1 \leq r \leq 255$  и обязательно  $r \leq m$ . По умолчанию  $r=0$ .  
Хотя применение типа `VARCHAR` экономит пространство и на диске, это не влияет на размер индекса, так как каждый ключ имеет длину равную  $m$ .

## Преобразования данных

### Преобразование число-число

	SMALLINT	INTEGER	SMALLFLOAT	FLOAT	DECIMAL
SMALLINT	ok	ok	ok	ok	0
INTEGER	X	ok	x	ok	0
SMALLFLOAT	X	X	F	ok	0
FLOAT	X	X	F	ok	0
DECIMAL	X	X	F	F	0

ok = без ошибок

0 = может быть ошибка, в зависимости от точности DECIMAL

X = может быть ошибка, в зависимости от данных

F = без ошибок, но младшие значащие цифры могут быть потеряны

В процессе функционирования системы может возникнуть необходимость преобразования типов столбцов таблиц. Например, столбец был создан с типом **SMALLINT** и позднее возникла необходимость хранить целые числа больше 32768. Преобразования типов столбцов выполняется предложением **ALTER TABLE**.

При изменении типа данных, новый тип данных должен обеспечивать хранение имеющихся ранее данных.

В таблице приведены возможные преобразования число-число. Например, при преобразовании числа типа `FLOAT` в `DECIMAL(4,2)` система округляет число перед записью в виде десятичного. При этом может возникнуть ошибка переполнения нового поля.

Можно проводить преобразования число-символьная строка и обратно. Однако, если строка содержит символы, которые недопустимы в числе (например символ `e` вместо `1`), то система не сможет выполнить данное преобразование.



## Преобразование DATE-DATETIME-INTERVAL

Тип данных операнда 1	Оператор	Тип данных операнда 2	Результат
DATE	-	DATETIME	INTERVAL
DATETIME	-	DATE	INTERVAL
DATE	+or-	INTERVAL	DATETIME
DATETIME	-	DATETIME	INTERVAL
DATETIME	+or-	INTERVAL	DATETIME
INTERVAL	+	DATETIME	DATETIME
INTERVAL	+or-	INTERVAL	INTERVAL
DATETIME	-	CURRENT	INTERVAL
CURRENT	-	DATETIME	INTERVAL
INTERVAL	+	CURRENT	DATETIME
CURRENT	+or-	INTERVAL	DATETIME
DATETIME	+or-	UNITS	DATETIME
INTERVAL	+or-	UNITS	INTERVAL
INTERVAL	*or/	NUMBER	INTERVAL

CURRENT - текущее время, UNITS - ключевое слово, задающее простой интервал (в единицах какого -либо поля времени).