

SQL



ЧАСТЬ 7.

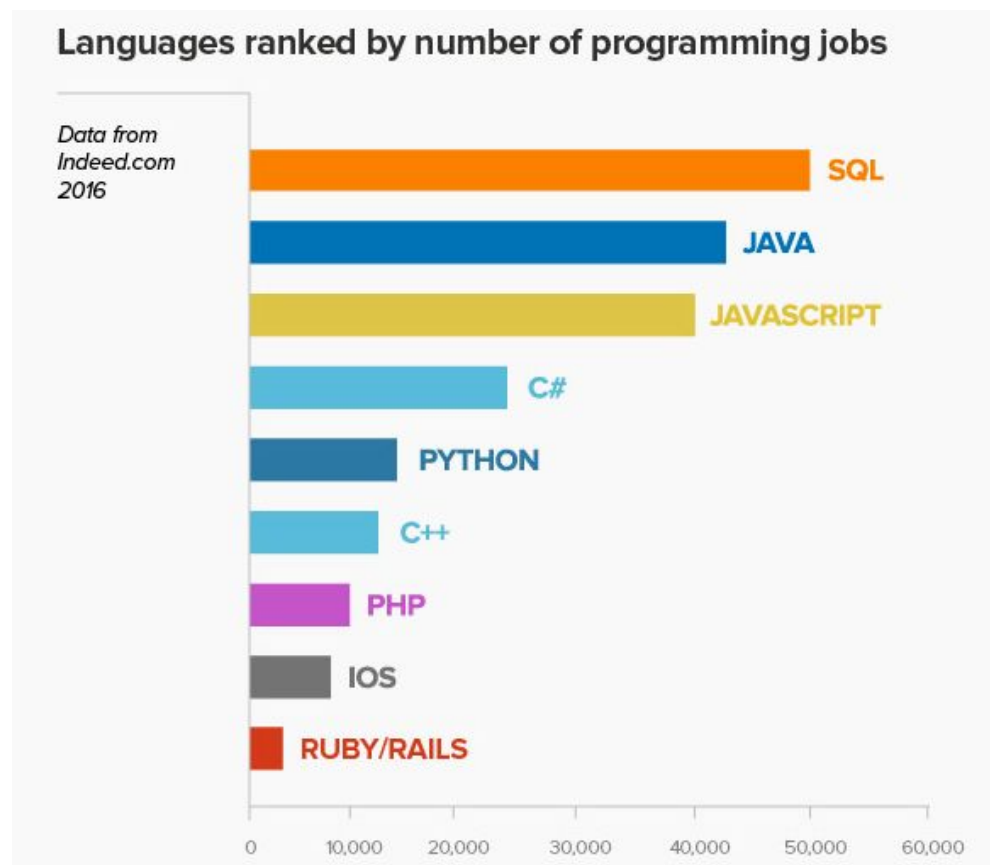
SQL

SQL (*Structured Query Language* — язык структурированных запросов) — универсальный компьютерный язык, применяемый для создания, модификации и управления данными в реляционных базах данных. Является информационно-логическим языком, а не языком программирования.



SQL

Основан на реляционном исчислении



ИСТОРИЯ

В начале 1970-х годов в компании IBM была разработана экспериментальная СУБД «System R» на основе языка *SEQUEL (Structured English Query Language* — структурированный английский язык запросов).

ИСТОРИЯ

В 1981 году IBM объявила о своём первом основанном на SQL программном продукте — SQL/DS.

в 1986 году первый стандарт языка SQL был принят ANSI (American National Standards Institute)

в 1987 году первый стандарт языка SQL был принят ISO (Международной организацией по стандартизации) в (SQL level 1) и уточнён в 1989 году (SQL level 2).

ИСТОРИЯ

1992 г. - новый расширенный стандарт (ANSI SQL-92 или SQL-2).

1999 г. - следующий стандарт SQL-99

2003 г. - SQL-3

2008 г. - действует в настоящее время

ПРЕИМУЩЕСТВА

- **Независимость от конкретной СУБД**
- **Наличие стандартов**
- **Полноценность как языка для управления данными**

НЕДОСТАТКИ

- **Нереляционность**
- **Сложность**
- **Отступления от стандартов**
- **Сложность работы с иерархическими структурами**

ЯЗЫК **SQL** ДЕЛИТСЯ НА ТРИ ЧАСТИ:

операторы определения данных (*Data Definition Language*, DDL)

операторы манипуляции данными (*Data Manipulation Language*, DML)

операторы определения доступа к данным (*Data Control Language*, DCL)

МЕТОДЫ ИСПОЛЬЗОВАНИЯ **SQL**

Статический (в тексте программы имеются вызовы функций языка SQL, которые жестко включаются в выполняемый модуль после компиляции)

Динамический (динамическое построение вызовов SQL-функций и интерпретация этих вызовов)

ПРЕДСТАВЛЕНИЕ -

таблица, формируемая в результате выполнения запроса.

КУРСОР

указатель, используемый для перемещения по наборам записей при их обработке.

В описательной части – связывание переменной типа CURSOR с оператором SQL; в выполняемой – открытие курсора (OPEN), перемещение (FETCH), сопровождение обработкой и закрытие (CLOSE).

ОПЕРАТОРЫ SQL

Арифметические операторы SQL

Оператор	Описание
+	Добавление
-	Вычитание
*	Умножение
/	Деление
%	Модуль

Побитовые операции SQL

Оператор	Описание
&	Побитовое И
	Побитовое ИЛИ
^	Побитовое исключение ИЛИ

ОПЕРАТОРЫ SQL

Операторы сравнения SQL

Оператор	Описание
=	Равно
>	Больше
<	Меньше
>=	Больше или равно
<=	Меньше или равно
<>	Не равно

ОПЕРАТОРЫ SQL

Составные операторы SQL

Оператор	Описание
+ =	Добавить равно
- =	Вычесть равно
*=	Умножить равно
/=	Разделить равно
%=	Модуль равно
&=	Побитовое И равно
^ - =	Побитовое исключающее равно
* =	Побитовое ИЛИ равно

ОПЕРАТОРЫ SQL

Логические операторы SQL

Оператор	Описание
ALL	Если все значения подзапроса являются TRUE
AND	Если все условия, разделенные И, являются TRUE
ANY	Если какое-либо из значений подзапроса соответствует TRUE условию
BETWEEN	Если операнд находится в диапазоне сравнения
EXISTS	Если подзапрос возвращает одну или несколько записей
IN	Если операнд равен одному из списка выражений
LIKE	Если операнд соответствует шаблону
NOT	Отображает запись, если условие (И) НЕ TRUE
OR	Если любое из условий, разделенных OR, является TRUE.
SOME	Если какое-либо из значений подзапроса соответствует условию

КОМАНДЫ ЯЗЫКА **DDL**

**CREATE TABLE,
ALTER TABLE,
DROP TABLE,
CREATE INDEX,
ALTER INDEX,
DROP INDEX.**

КОМАНДЫ ЯЗЫКА **DML**

INSERT,

UPDATE,

DELETE.

SELECT

Выборка данных (DQL)

КОМАНДЫ ЯЗЫКА **DCL**

GRANT

REVOKE

КОМАНДЫ УПРАВЛЕНИЯ ТРАНЗАКЦИЯМИ

**COMMIT,
ROLLBACK,
SAVEPOINT,
SET TRANSACTION.**

ЗАПИСЬ SQL- ОПЕРАТОРОВ

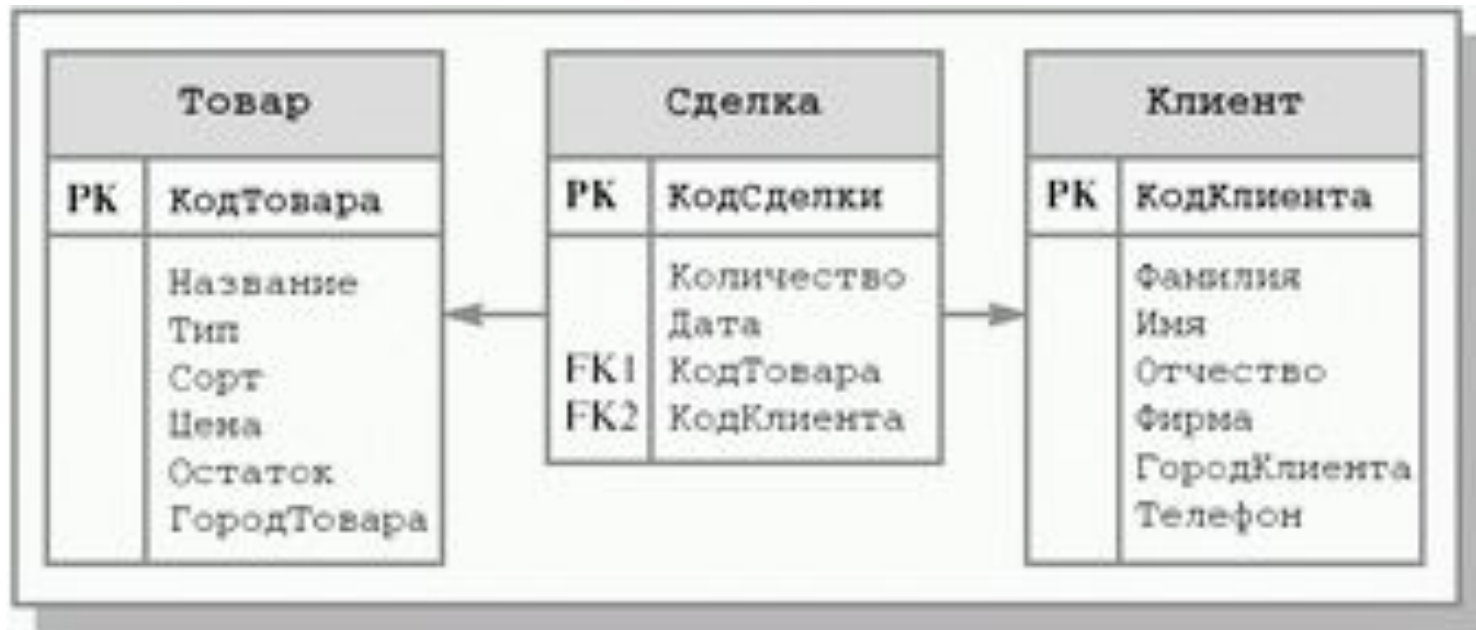
**<идентификатор> ::= <буква>
{<буква>|<цифра>}[...n]**

**идентификатор может иметь длину до
128 символов;**

**идентификатор должен начинаться с
буквы;**

**идентификатор не может содержать
пробелы.**

БАЗА ДАННЫХ



ТИПЫ ДАННЫХ ЯЗЫКА **SQL**, ОПРЕДЕЛЕННЫЕ СТАНДАРТОМ

Символьный CHAR | VARCHAR

Битовый BIT | BIT VARYING

Точные числа NUMERIC | DECIMAL | INTEGER |
SMALLINT

Округленные числа FLOAT | REAL | DOUBLE
PRECISION

Дата/время DATE | TIME | TIMESTAMP

Интервал INTERVAL

ЧИСЛОВЫЕ ТИПЫ

Тип	Описание
TINYINT[(max)]	Очень маленькие целые числа диапазона -127... 128.
SMALLINT[(max)]	Маленькие целые числа диапазона -32768... 32767.
MEDIUMINT[(max)]	Средние целые числа.
INT[(max)]	Обычные целые числа.
FLOAT[(max,P)]	Числа с плавающей точкой одинарной точности.
DOUBLE[(max,P)]	Числа с плавающей точкой двойной точности.
DECIMAL[(max,P)]	Числа с плавающей точкой, приведенные к типу char.

ТИПЫ ДАТЫ И ВРЕМЕНИ

Тип	Описание
DATE	Дата в формате ГГГГ-ММ-ДД.
TIME	Время в формате ЧЧ-ММ-СС.
DATETIME	Дата и время в формате ГГГГ-ММ-ДД ЧЧ-ММ-СС.
YEAR	Год в формате ГГ или ГГГГ.
TIMESTAMP	Метка времени для отсчетов по транзакциям в формате ГГГГ-ММ-ДД ЧЧ-ММ-СС.

СТРОКОВЫЕ ТИПЫ

Тип
CHAR (len)[BINARY]
CHAR
VARCHAR (len)[BINARY]
TEXT
BLOB

Тип	Описание
CHAR (len)[BINARY]	Строки с длиной len, которое не превышает 255 символов. Ключевое слово BINARY указывает на то, что данные должны обрабатываться независимо от регистра.
CHAR	Синоним CHAR(1).
VARCHAR (len)[BINARY]	Синоним CHAR(len) за исключением того, что строки могут быть произвольной длины.
TEXT	Строки с максимальной длиной символов равной 65535. Данные этого типа чувствительны к регистру.
BLOB	Двоичные строки с максимальной длиной символов равной 65535. Тип BLOB (binary large object - большой двоичный объект) предназначен для хранения двоичных данных, в том числе изображений и звуковых последовательностей.

ДОМЕН –

это набор допустимых значений для одного или нескольких атрибутов.

```
<определение_домена> ::= CREATE  
DOMAIN имя_домена [AS] тип_данных [  
DEFAULT значение] [ CHECK  
(допустимые_значения)]
```

```
DROP DOMAIN имя_домена [ RESTRICT |  
CASCADE]
```

ТИПЫ ДАННЫХ, ИСПОЛЬЗУЕМЫЕ В **SQL-СЕРВЕРЕ**

image

smalldatetime

bit

text

real

char

uniqueidentifier

numeric

timestamp

datetime

binary

ntext

decimal

nchar

money

tinyint

Smallmoney

Nvarchar

smallint

float

varbinary

int

varchar

sysname

СОЗДАНИЕ ПОЛЬЗОВАТЕЛЬСКОГО ТИПА ДАННЫХ

sp_addtype

[@typename=]type,[@phystype=]

system_data_type [,[@nulltype=]'null_type']

EXEC sp_addtype bir, DATETIME, 'NULL'

ПРЕОБРАЗОВАНИЕ ТИПОВ

CAST(выражение AS тип_данных)

**CONVERT(тип_данных[(длина)],
выражение [,стиль])**

STR

ПЕРЕМЕННЫЕ

```
DECLARE {@имя_переменной тип_данных } [,...n]
```

```
DECLARE @a INT
```

```
    SET @a=10
```

```
DECLARE @k INT
```

```
    SELECT @k=SUM(количество) FROM  
        Товар
```


УПРАВЛЯЮЩИЕ КОНСТРУКЦИИ **SQL**

```
<блок_операторов> ::= BEGIN  
    { sql_оператор | блок_операторов }  
    END
```

УСЛОВНЫЙ ОПЕРАТОР

<условный_оператор>::=

IF лог_выражение

{ sql_оператор | блок_операторов }

[ELSE {sql_оператор | блок_операторов }

]

ОПЕРАТОР ВЫБОРА ВАРИАНТА

```
<оператор_выбора_варианта> ::=  
    CASE входное_значение  
    WHEN {значение_для_сравнения |  
лог_выражение }  
        THEN вых_выражение [,...n]  
        [ ELSE иначе_вых_значение ]  
    END
```

ЦИКЛЫ

<оператор_цикла> ::=

WHILE лог_выражение

{ sql_оператор | блок_операторов }

[BREAK]

{ sql_оператор | блок_операторов }

[CONTINUE]

ОСНОВНЫЕ ОБЪЕКТЫ СТРУКТУРЫ БАЗЫ ДАННЫХ **SQL-** СЕРВЕРА

Tables

Views

Stored Procedures

Triggers

User Defined function

Indexes

User Defined Data Types

Keys

Constraints

Users

Roles

Rules

Defaults

СОЗДАНИЕ БАЗЫ ДАННЫХ В СРЕДЕ **MS SQL SERVER**

<определение_базы_данных> ::=

CREATE DATABASE имя_базы_данных

[ON [PRIMARY] [<определение_файла> [,...n]]

[,<определение_группы> [,...n]]] [LOG ON

{<определение_файла>[,...n] }]

[FOR LOAD | FOR ATTACH]

ОПРЕДЕЛЕНИЕ ФАЙЛА

**<определение_файла>::=
([NAME=логическое_имя_файла,]
FILENAME='физическое_имя_файла'
[, SIZE=размер_файла]
[, MAXSIZE={max_размер_файла | UNLIMITED }]
[, FILEGROWTH=величина_прироста]) [, ...n]**

ИЗМЕНЕНИЕ БАЗЫ ДАННЫХ

<изменение_базы_данных> ::=

ALTER DATABASE имя_базы_данных

{ ADD FILE <определение_файла>[,...n]

[TO FILEGROUP имя_группы_файлов]

| ADD LOG FILE <определение_файла>[,...n]

| REMOVE FILE логическое_имя_файла

| ADD FILEGROUP имя_группы_файлов

| REMOVE FILEGROUP имя_группы_файлов

| MODIFY FILE <определение_файла>

| MODIFY FILEGROUP имя_группы_файлов

<свойства_группы_файлов>}

УДАЛЕНИЕ БАЗЫ ДАННЫХ

DROP DATABASE имя_базы_данных [,...n]

СОЗДАНИЕ ТАБЛИЦЫ

<определение_таблицы> ::=

**CREATE TABLE имя_таблицы
(имя_столбца тип_данных
[NULL | NOT NULL] [,...n])**

ПРИМЕР

CREATE TABLE Товар

**(Название VARCHAR(50) NOT NULL, Цена
MONEY NOT NULL,**

Тип VARCHAR(50) NOT NULL,

Сорт VARCHAR(50),

Город_Товара VARCHAR(50))

ИЗМЕНЕНИЕ ТАБЛИЦЫ

ALTER TABLE имя_таблицы

**{[ALTER COLUMN имя_столбца
{новый_тип_данных [(точность[,масштаб)))] [NULL | NOT NULL]}}**

| ADD { [имя_столбца тип_данных]

| имя_столбца AS выражение } [,...n]

| DROP {COLUMN имя_столбца}[,...n] }

ALTER TABLE Клиент ADD Рас_счет CHAR(20)

ПРАВИЛА ПРИ ИЗМЕНЕНИИ ОПРЕДЕЛЕНИЙ СТОЛБЦОВ

- **размер столбца может быть увеличен до максимального значения, допускаемого соответствующим типом данных;**
- **размер столбца может быть уменьшен только в том случае, если содержащееся в нем наибольшее значение не будет превосходить его нового размера;**
- **количество разрядов числового типа данных всегда может быть увеличено;**
- **количество разрядов числового типа данных может быть уменьшено только в том случае, если количество разрядов наибольшего значения в соответствующем столбце не будет превосходить нового числа разрядов, определенного для этого столбца;**
- **количество десятичных знаков числового типа данных может быть уменьшено или увеличено;**
- **тип данных столбца, как правило, может быть изменен.**