

Язык баз данных SQL

Предложен: 1974 г., исследовательский проект IBM System R

Первоначальное название: SEQUEL (Structured English QUery Language)

Стандарт: ANSI, ISO (ISO-9075)

SQL-ориентированные коммерческие СУБД:

1979 г. Oracle

1981 г. IBM SQL/DS (до конца 90-х)

1983 г. IBM DB2

1985 г. INFORMIX SQL (в 2001 г. приобретена IBM)

1987 г. Sybase SQL Server

1989 г. Microsoft SQL Server (по технологии Sybase)

Наиболее распространенные некоммерческие реализации:

MySQL (с 2010 г. принадлежит Oracle)

PostgreSQL

Возможности языка SQL

- Формулирование запросов к БД
- Манипулирование данными (создание, модификация, удаление)
- Определение и манипулирование схемой БД
- Определение ограничений целостности данных
- Определение представлений (виртуальных таблиц)
- Определение структур физического уровня, поддерживающих эффективное исполнение запросов
- Авторизация доступа к данным
- Управление транзакциями

Стандартизация языка SQL

Поко- ление	Год	Стандарт	Основные изменения
1	1986	SQL-86	Первоначальная версия
	1989	SQL-89	Четкая стандартизация синтаксиса и семантики операторов выборки данных, манипулирования данными и определения ограничений целостности
2	1992	SQL-92	Манипулирование схемой, транзакциями, сессиями, подключениями, динамический SQL
3	1995	-	SQL/CLI (Call-Level Interface)
	1996	-	SQL/PSM (Persistent Stored Modules)
	1999	SQL:1999	ОО расширения, регулярные выражения, триггеры, рекурсивные запросы
	2003	SQL:2003	Java, XML, поддержка OLAP
	2006	-	Переработана поддержка XML (XQUERY)
	2008	SQL:2008	Улучшена поддержка OLAP, XQUERY, расширение триггеров
	2011	SQL:2011	Улучшена поддержка темпоральных данных
	2016	SQL:2016	JSON, полиморфные табличные функции, поиск строк по шаблону
	2019	-	Поддержка многомерных массивов

Структура стандарта SQL (2003 - 2016)

Том 1: Framework

Концептуальная структура стандарта

Том 2: Foundation

Синтаксис, семантика, правила связывания для процедурных языков программирования

Том 3: Call-Level Interface

Интерфейс уровня вызовов (основа SQL-ориентированных API)

Том 4: Persistent Stored Modules

Описание языка SQL/PSM

Том 9: Management of External Data

Языковые средства взаимодействия с внешними данными

Том 10: Object Language Bindings

Правила связывания для объектно-ориентированных языков программирования

Том 11: Information and Definition Schemas

Описание информационной схемы (хранение описателей данных)

Том 13: SQL Routines and Types Using the Java Programming Language

Использование SQL совместно с языком программирования Java

Том 14: XML-Related Specifications

Языковые средства работы с XML документами

Том 15 (2019): Multi-dimensional Arrays

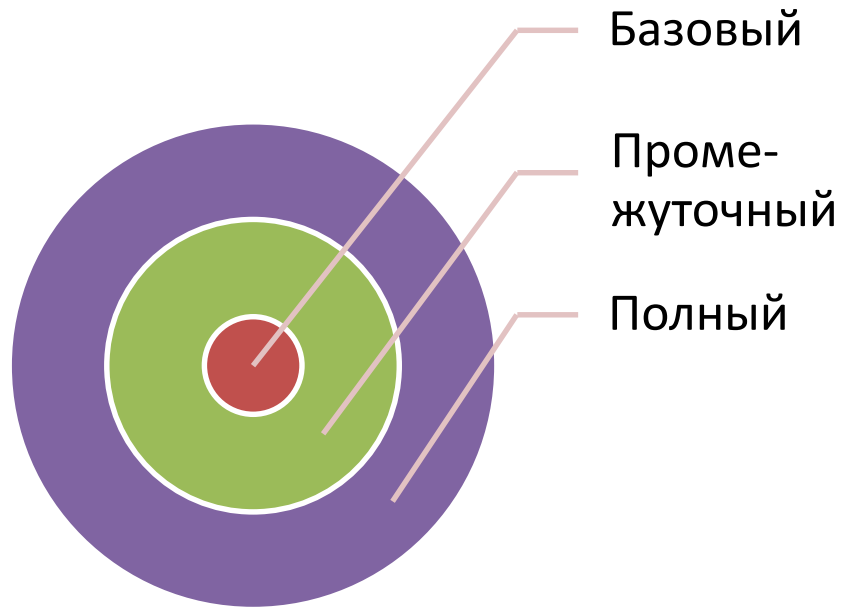
Поддержка работы с многомерными массивами

Критика языка SQL

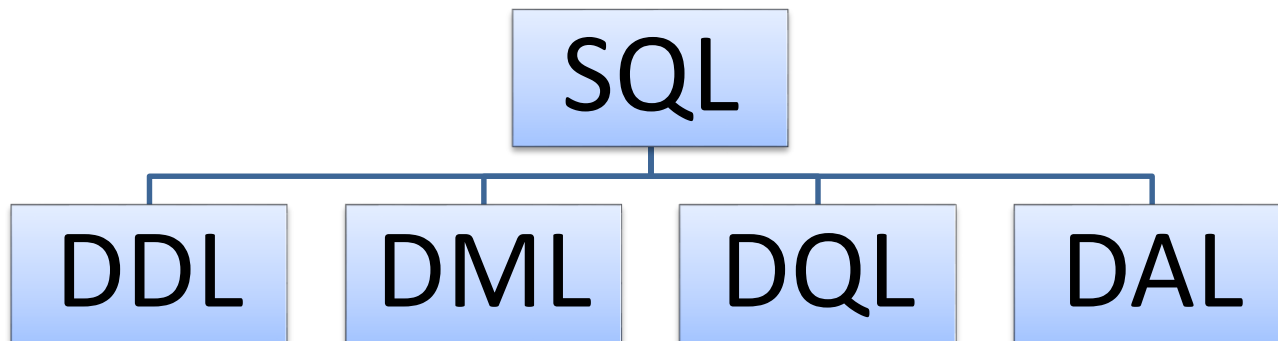
1. Противоречие принципам реляционной модели данных
 - Терминология (РМД – отношение, атрибут, кортеж; SQL – таблица, столбец, строка)
 - Отношение РМД – множество кортежей; таблица SQL – мультимножество строк
 - Заголовок таблицы SQL – упорядоченное множество столбцов (в РМД – множество атрибутов)
 - Во временных таблицах, порождаемых запросами SQL, допускаются безымянные столбцы, а также дублирование имен столбцов
 - В SQL значения NULL допускаются для возможных ключей, существует несколько способов сопоставления внешнего и возможного ключей
2. Чрезмерная избыточность языка
3. Отступления от стандарта в реализациях (диалекты SQL)

Структура языка SQL

С точки зрения
разработчиков СУБД



С точки зрения программиста
приложений БД



Условное
разделение на
подъязыки

Типы данных SQL

- Данные, хранящиеся в таблицах SQL, являются типизированными
- СУБД должна отслеживать, чтобы в каждом столбце каждой строки таблицы присутствовали только допустимые для соответствующих типов данных значения
- NULL является допустимым значением для любого типа данных SQL

Булевский тип:

BOOLEAN

Литералы: FALSE, TRUE, UNKNOWN

SQL:2003: «В этой спецификации не проводится различие между NULL-значением булевского типа данных и истинностным значением UNKNOWN, являющимся результатом вычисления предиката, условия поиска или булевского выражения. Они могут использоваться взаимозаменяемо и означают в точности одно и то же»

Числовые типы данных

Точные числовые типы

Целые числа:

SMALLINT, INTEGER, BIGINT

точность представления определяется в реализации

Числа с фиксированной точкой:

NUMERIC(p, s), DECIMAL(p, s)

p – точность (число сохраняемых десятичных цифр): строгая для NUMERIC, минимально допустимая для DECIMAL

s – масштаб (число десятичных цифр в дробной части)

допустимые значения p, s определяются в реализации ($p \geq s$)

значения по умолчанию: $s=0$, p – определяется в реализации

Приближенные числовые типы

REAL, DOUBLE PRECISION, FLOAT(p)

p – число значащих цифр в мантиссе

точность представления для первых двух типов и максимальное значение

p определяются в реализации

Типы символьных и битовых строк

CHAR(x), VARCHAR(x), CLOB(z)

x, z – количество символов в строке: фиксированное для CHAR, максимально допустимое для VARCHAR и CLOB
допустимые значения x, z определяются в реализации (z >> x),
z может указываться в виде nK, nM, nG
вид литералов: 'abcdef' или "FBCDE"
набор допустимых символов определяется в реализации
(ASCII как минимум)

BIT(x), BIT VARYING(x), BLOB(z)

в отличие от строк состоят из произвольных байтов, не
обязательно кодирующих символы
вид литералов: B'01001101101' или X'79CA83'

Типы даты, времени

DATE, TIME(p), TIMESTAMP(p), TIME WITH TIMEZONE(p),
TIMESTAMP WITH TIMEZONE(p)

p – точность представления долей секунды, максимальное значение определяется в реализации (≥ 6), значение по умолчанию p=0 для TIME, p=6 для TIMESTAMP

Допустимое значение секунд варьируется от 00 до 61

Форматы литералов: DATE 'yyyy-mm-dd', TIME 'hh:mm:ss:ff...f',
TIMESTAMP 'yyyy-mm-dd hh:mm:ss:ff...f'

Временная зона задается в виде +hh:mm или -hh:mm

Функции, возвращающие текущие дату и время: CURRENT_DATE,
CURRENT_TIME, CURRENT_TIMESTAMP

Типы временных интервалов

INTERVAL start(p) [TO end(q)]

start, end: YEAR, MONTH, DAY, HOUR, MINUTE, SECOND

Поддерживаемые интервалы:

YEAR, YEAR TO MONTH, MONTH

DAY, DAY TO HOUR, DAY TO MINUTE, DAY TO SECOND

HOUR, HOUR TO MINUTE, HOUR TO SECOND

MINUTE, MINUTE TO SECOND, SECOND

p – точность представления лидирующего поля, q – точность долей секунды (применяется, если end=SECOND), максимальные значения определяются в реализации, значения по умолчанию p=2, q=6

Формат литералов: INTERVAL '1:35' HOUR TO MINUTE

Преобразование типов данных в SQL

- Поддерживается явное и неявное преобразование значений одного типа данных к другому
- Правила неявного преобразования не сильно отличаются от тех, что существуют в языках программирования
 - Точный числовой тип приводится к другому точному числовому типу с большей точностью или к приближенному числовому типу
 - Приближенный числовой тип приводится к другому приближенному числовому типу с большей точностью
 - Строковый тип фиксированной длины приводится к другому строковому типу фиксированной или переменной длины с большим допустимым количеством символов
 - Строковый тип переменной длины приводится к другому строковому типу переменной длины с большим допустимым количеством символов
- Явное преобразование осуществляется с помощью оператора CAST(expression AS datatype)
- Подробные правила выполнения оператора CAST см. в учебной литературе

Домены в SQL

Домен – базовый тип данных + ограничение области определения значений
Домен является долговременно хранимым именованным объектом схемы БД

Оператор определения домена:

```
CREATE DOMAIN domain_name AS datatype [DEFAULT value] [constraint_list]
```

DEFAULT value – определение значения по умолчанию (литерал базового типа, либо значение NULL, либо вызов предопределенной функции, возвращающей литеральное значение базового типа)

constraint_list – список ограничений области определения значений, все ограничения в списке связываются логической функцией AND

Пример:

```
CREATE DOMAIN SALARY AS NUMERIC( 10, 2 )  
    DEFAULT 10000.00  
    CHECK( VALUE BETWEEN 10000.00 AND 100000000.00 )  
    CONSTRAINT SAL_NOT_NULL CHECK( VALUE IS NOT NULL );
```

Домены в SQL

Оператор изменения определения домена:

ALTER DOMAIN domain_name action

Допустимы 4 действия: SET DEFAULT value, DROP DEFAULT, ADD CONSTRAINT [constraint_name] constraint_expression, DROP CONSTRAINT constraint_name

Если к моменту исполнения ALTER DOMAIN ADD CONSTRAINT существуют столбцы таблиц, определенные на данном домене, текущие значения которых противоречат новому ограничению, то СУБД должна отвергнуть этот оператор

Отменить можно только именованное ограничение

Пример:

```
ALTER DOMAIN SALARY SET DEFAULT 15000.00;
```

```
ALTER DOMAIN SALARY DROP CONSTRAINT SAL_NOT_NULL;
```

Домены в SQL

Оператор отмены определения домена:

`DROP DOMAIN domain_name { RESTRICT | CASCADE }`

`DROP DOMAIN RESTRICT` отвергается, если домен использован в определении некоторого столбца таблицы (базовой или виртуальной) или в определении ограничения целостности

`DROP DOMAIN CASCADE` выполняется всегда по следующим правилам:

- Уничтожаются все ограничения целостности и виртуальные таблицы, в определениях которых задействован данный домен
- Столбцы базовых таблиц, определенные на данном домене, преобразуются к его базовому типу и наследуют значение по умолчанию и все ограничения уничтожаемого домена

Таблицы в SQL

1. Базовые – реально хранимые в БД таблицы
2. Порождаемые – таблицы, формируемые и существующие во время выполнения запросов
3. Представления (виртуальные) – таблицы, существование которых поддерживается алгоритмически

Оператор определения базовой таблицы:

```
CREATE TABLE table_name ( column_list [constraint_list] )  
column_definition ::= column_name { datatype | domain } [DEFAULT  
value] [constraint_list]
```

Данный оператор создает соответствующие описатели в схеме БД и выделяет область во внешней памяти для хранения данных.

Значение по умолчанию для столбца

Действующее значение по умолчанию для столбца определяется следующим образом:

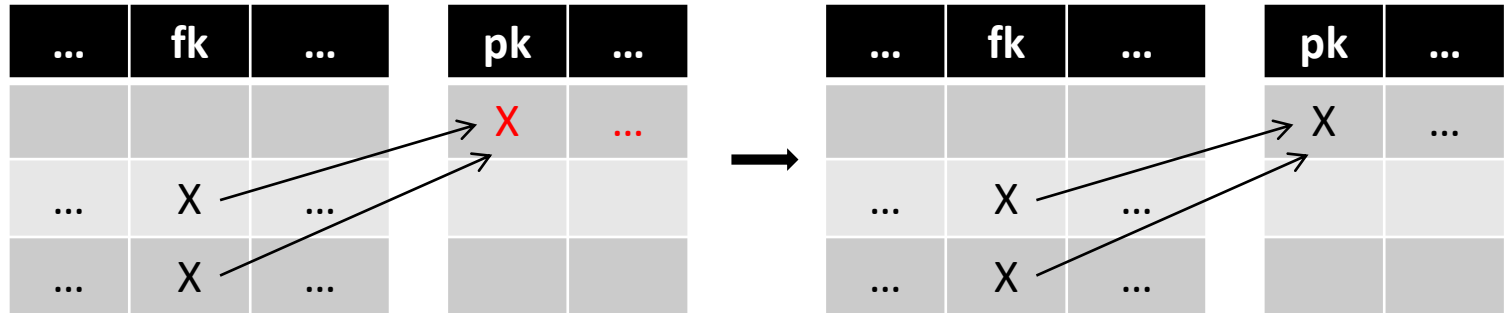
1. Если в определении столбца явно присутствует раздел DEFAULT, то действует значение указанное в данном разделе
2. Иначе, если столбец определен на домене и в определении этого домена явно присутствует раздел DEFAULT, то действует значение, указанное в данном разделе
3. Иначе, если значение NULL не является запрещенным для данного столбца, то оно является значением по умолчанию
4. Иначе, значение по умолчанию у данного столбца отсутствует (в этом случае при любой вставке новой строки в данную таблицу значение для данного столбца должно быть явно задано)

Ограничения целостности столбцов и таблиц

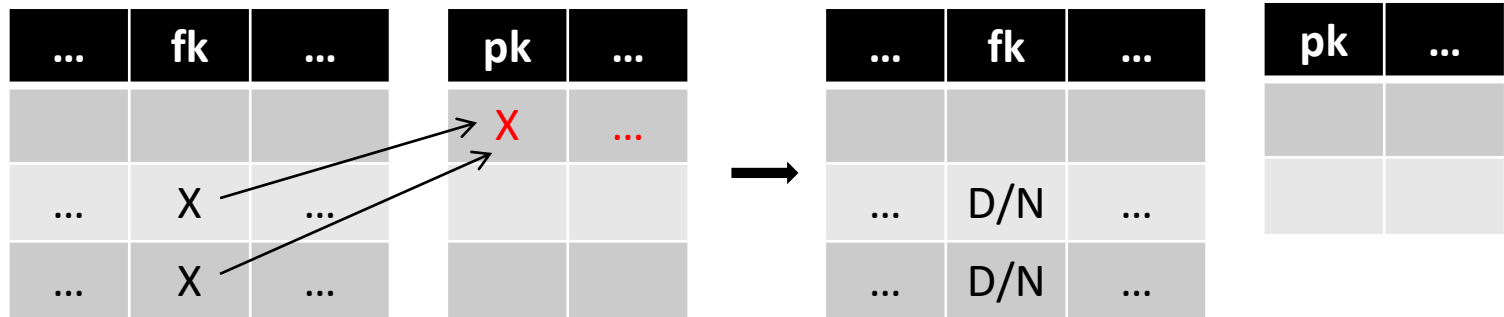
1. NOT NULL (только для столбцов, эквивалентное табличное ограничение для столбца C: CHECK(C IS NOT NULL)
2. Проверочное ограничение: CHECK(logical_expression), попытка обновления таблицы нарушает проверочное ограничение в том и только том случае, если результат вычисления логического выражения равен FALSE
3. Ограничение первичного ключа: PRIMARY KEY(column_list), допускается не более одного определения первичного ключа в таблице, PRIMARY KEY подразумевает NOT NULL для каждого из столбцов, упоминаемых в определении первичного ключа
4. Ограничение возможного ключа: UNIQUE(column_list), NULL допускается, но может встретиться только один раз в столбце возможного ключа
5. Ограничение внешнего ключа:
для столбца REFERENCES foreign_table_name [(foreign_table_column)] [ON DELETE action] [ON UPDATE action]
для таблицы FOREIGN KEY column_list REFERENCES foreign_table_name [(foreign_table_column_list)] [ON DELETE action] [ON UPDATE action]

Ссылочные действия (на примере DELETE)

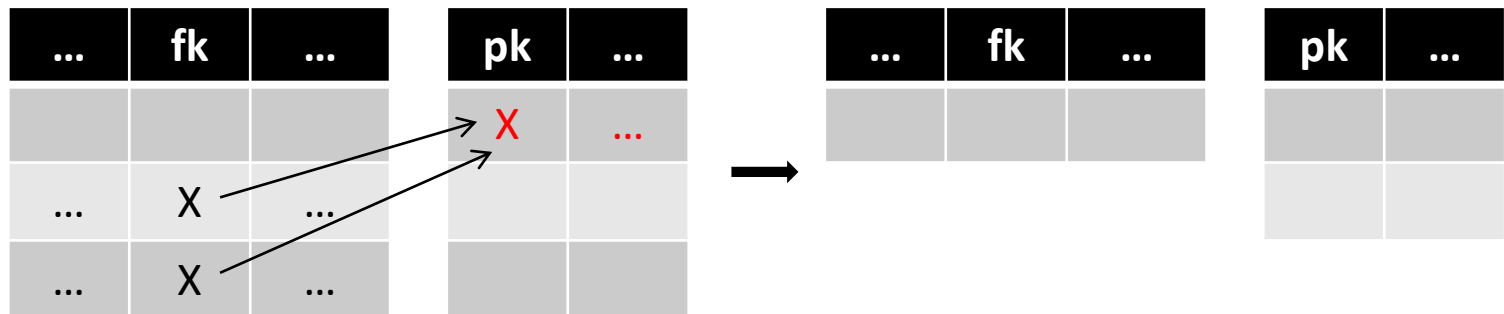
NO ACTION
или
RESTRICT



SET DEFAULT
или
SET NULL

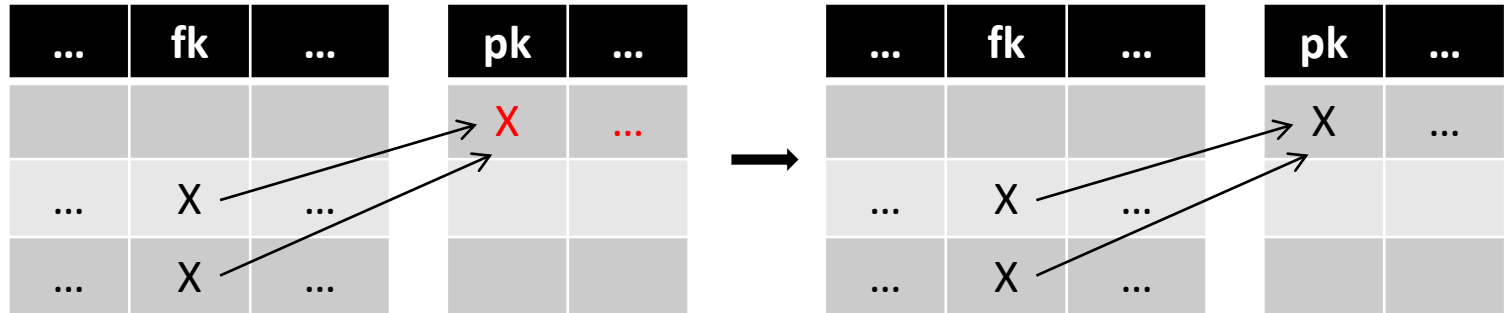


CASCADE

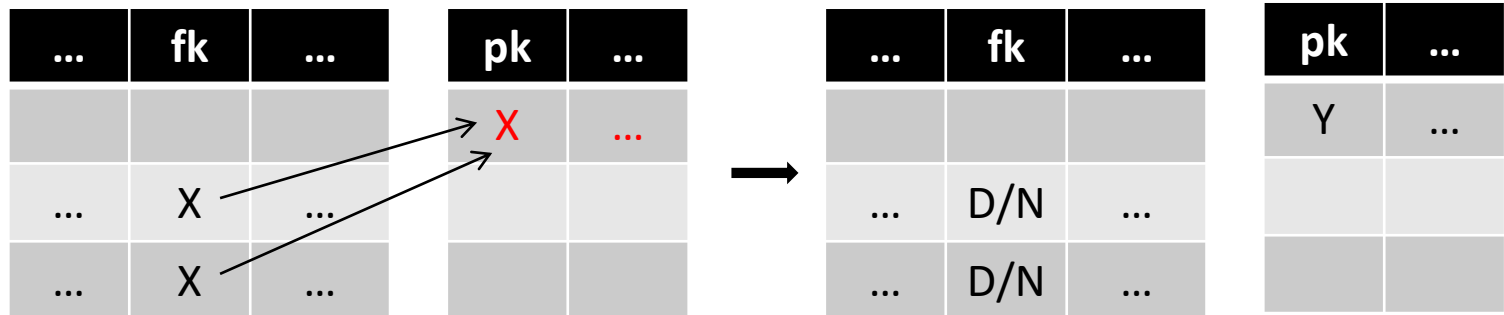


Ссылочные действия (на примере UPDATE)

NO ACTION
или
RESTRICT



SET DEFAULT
или
SET NULL



CASCADE

