

# Дополнительные главы баз данных (Модель данных SQL)

Сергей Дмитриевич Кузнецов  
Институт системного программирования РАН  
kuzloc@ispras.ru

# Содержание курса (1)

- История стандарта языка SQL. Типы данных. Средства языка SQL для определения и изменения доменов, базовых таблиц и ограничений целостности
  - История стандарта SQL и структура языка
    - ✓ Этапы процесса стандартизации языка SQL
    - ✓ Структура языка SQL

# Содержание курса (2)

## ➤ Типы данных SQL

- ✓ Точные числовые типы, приближенные числовые типы, типы символьных и битовых строк, темпоральные типы, Булевский тип, типы коллекций, анонимные строчные типы, типы, определяемые пользователем, ссылочные типы
- Средства определения, изменения и ликвидации доменов
- Средства определения, изменения и ликвидации базовых таблиц
- Средства определения и отмены общих ограничений целостности

# Содержание курса (3)

- Базовые возможности выборки данных в языке SQL
  - Общая структура оператора выборки в языке SQL
    - ✓ Семантика оператора выборки
    - ✓ Ссылки на таблицы раздела FROM
      - Табличное выражение, спецификация запроса и выражение запросов
      - Ссылки на базовые, представляемые и порождаемые таблицы
  - Представляемые таблицы, или представления

# Содержание курса (4)

- **Базовые возможности выборки данных в языке SQL**
  - Логические выражения раздела WHERE
    - ✓ Предикат сравнения, предикат between, предикат null, предикат in, предикат like, предикат similar, предикат exists, предикат unique, предикат overlaps, предикат сравнения с квантором, предикат match, предикат distinct
  - Логические выражения раздела HAVING

# Содержание курса (5)

- Базовые возможности модификации баз данных в языке SQL
  - Представления, над которыми возможны операции обновления
    - ✓ Представления, допускающие применение операций обновления, в стандарте SQL/92
    - ✓ Представления, допускающие применение операций обновления, в стандарте SQL:1999
    - ✓ Раздел WITH CHECK OPTION определения представления
    - ✓ Исторический очерк

# Содержание курса (6)

- Базовые возможности модификации баз данных в языке SQL
  - Операции обновления баз данных и механизм триггеров
    - Понятие триггера в SQL:1999
    - Синтаксис определения триггеров и типы триггеров
    - Выполнение триггеров
    - Триггеры и ссылочные действия

# Содержание курса (7)

- Механизмы авторизации доступа и управления подключениями, сессиями и транзакциями в языке SQL
  - Поддержка авторизации доступа к данным в языке SQL
    - ✓ Пользователи и роли
    - ✓ Использование идентификаторов пользователей и имен ролей
    - ✓ Создание и ликвидация ролей
    - ✓ Передача привилегий и ролей
    - ✓ Изменение текущих идентификаторов пользователей и имен ролей
    - ✓ Аннулирование привилегий и ролей



# Содержание курса (8)

- Механизмы авторизации доступа и управления подключениями, сессиями и транзакциями в языке SQL
  - Управление транзакциями в SQL
    - ✓ Порождение транзакций в SQL
    - ✓ Уровни изоляции SQL-транзакции
    - ✓ Завершение транзакций
    - ✓ Транзакции и ограничения целостности
    - ✓ Точки сохранения

# Содержание курса (9)

- Механизмы авторизации доступа и управления подключениями, сессиями и транзакциями в языке SQL

- Подключения и сессии

- ✓ Установление соединений
- ✓ Операторы SQL для управления соединениями
  - Оператор CONNECT
  - Оператор SET CONNECTION
  - Оператор DISCONNECT

# Лекция 1. Общее введение в SQL

- В начале лекции мы представим небольшой исторический обзор SQL
  - язык уже далеко не молод
  - в 2024 г. сообщество баз данных отметит его 50-летний юбилей
  - чтобы правильно понимать и трактовать современные варианты SQL, нужно знать историю языка хотя бы в общих чертах

# Краткая история языка SQL (1)

- Язык SQL, предназначенный для взаимодействия с базами данных, появился в середине 70-х гг. (первые публикации датируются 1974 г.) и был разработан в компании IBM в рамках проекта экспериментальной реляционной СУБД System R
- Исходное название языка SEQUEL (Structured English Query Language) только частично отражало суть этого языка
- Конечно, язык был ориентирован главным образом на удобную и понятную пользователям формулировку запросов к реляционным БД.

# Краткая история языка SQL (2)

- Но, в действительности, он почти с самого начала являлся полным языком БД, обеспечивающим помимо средств формулирования запросов и манипулирования БД следующие возможности:
  - средства определения и манипулирования схемой БД;
  - средства определения ограничений целостности и триггеров;
  - средства определения представлений БД;
  - средства определения структур физического уровня, поддерживающих эффективное выполнение запросов;
  - средства авторизации доступа к отношениям и их полям<sup>67)</sup>;
  - средства определения точек сохранения транзакции и выполнения фиксации и откатов транзакций

# Краткая история языка SQL (3)

- В языке отсутствовали средства явной синхронизации доступа к объектам БД со стороны параллельно выполняемых транзакций: с самого начала предполагалось, что необходимую синхронизацию неявно выполняет СУБД.
- В настоящее время язык SQL реализован во всех коммерческих реляционных СУБД и почти во всех СУБД, которые изначально основывались не на реляционном подходе
- Все компании-производители провозглашают соответствие своей реализации стандарту SQL, и на самом деле реализованные диалекты SQL очень близки
- Этого удалось добиться не сразу.

# Краткая история языка SQL (4)

- Наиболее близки к System R были две системы компании IBM – SQL/DS и DB2
- Разработчики обеих систем использовали опыт проекта System R, а СУБД SQL/DS напрямую основывалась на программном коде System R
- Отсюда предельная близость диалектов SQL, реализованных в этих системах, к SQL System R
- Из SQL System R были удалены только те части, которые были недостаточно проработаны (например, точки сохранения) или реализация которых вызывала слишком большие технические трудности (например, ограничения целостности и триггеры)
- Можно назвать этот путь к коммерческой реализации SQL движением сверху вниз.

# Краткая история языка SQL (5)

- Другой подход применялся в таких системах, как Oracle, Informix и Sybase. Несмотря на различие в способах разработки систем, реализация SQL везде происходила «снизу вверх»
- В первых выпущенных на рынок версиях этих систем использовалось ограниченное подмножество SQL System R
- В частности, в первой известной нам реализации SQL в СУБД Oracle в операторах выборки не допускалось использование вложенных подзапросов и отсутствовала возможность формулировки запросов с соединениями нескольких отношений



# Краткая история языка SQL (6)

- Особенностью большинства современных коммерческих СУБД, затрудняющей сравнение существующих диалектов SQL, является отсутствие единообразного описания языка
- Обычно описание разбросано по разным руководствам и перемешано с описанием специфических для данной системы языковых средств, не имеющих прямого отношения к SQL
- Тем не менее, можно сказать, что базовый набор операторов SQL, включающий операторы определения схемы БД, выборки и манипулирования данными, авторизации доступа к данным, поддержки встраивания SQL в языки программирования и операторы динамического SQL, в коммерческих реализациях устоялся и более или менее соответствует стандарту

# Краткая история языка SQL (7)

- Деятельность по стандартизации языка SQL началась практически одновременно с появлением его первых коммерческих реализаций
- В 1982 г. комитету по базам данных Американского национального института стандартов (ANSI) было поручено разработать спецификацию стандартного языка реляционных баз данных
- Первый документ из числа имеющихся у автора проектов стандарта датирован октябрём 1985 г. и является уже не первым проектом стандарта ANSI
- Стандарт был принят ANSI в 1986 г., а в 1987 г. одобрен Международной организацией по стандартизации (ISO)
- Этот стандарт принято называть *SQL/86*

# Краткая история языка SQL (8)

- Понятно, что в качестве основы стандарта нельзя было использовать SQL System R
- Во-первых, этот вариант языка не был должным образом технически проработан
- Во-вторых, его слишком сложно было бы реализовать (кто знает, как бы сложилась судьба SQL, если бы все идеи проекта System R были реализованы полностью)
- Поэтому за основу был взят диалект языка SQL, сложившийся в IBM к началу 1980-х гг.
- В сущности, этот диалект представлял собой технически проработанное подмножество SQL System R

# Краткая история языка SQL (9)

- К 1989 г. стандарт SQL/86 был несколько расширен, и был подготовлен и принят следующий стандарт, получивший название ANSI/ISO SQL/89
- Анализ доступных документов показывает, что процесс стандартизации SQL происходил очень сложно с использованием не только научных доводов
- В результате SQL/89 во многих частях имеет чрезвычайно общий характер и допускает очень широкое толкование
- В этом стандарте полностью отсутствуют такие важные разделы, как манипулирование схемой БД и динамический SQL
- Многие важные аспекты языка в соответствии со стандартом определяются в реализации

# Краткая история языка SQL (10)

- Наиболее важными достижениями стандарта SQL/89 являются четкая стандартизация синтаксиса и семантики операторов выборки данных и манипулирования данными и фиксация средств ограничения целостности БД
- Были специфицированы средства определения первичного и внешних ключей отношений и так называемых проверочных ограничений целостности, которые представляют собой подмножество немедленно проверяемых ограничений целостности SQL System R

# Краткая история языка SQL (11)

- Средства определения внешних ключей позволяют легко формулировать требования так называемой ссылочной целостности БД
- Это распространенное в реляционных БД требование можно было сформулировать и на основе общего механизма ограничений целостности SQL System R, но формулировка на основе понятия внешнего ключа более проста и понятна

# Краткая история языка SQL (12)

- Осознавая неполноту стандарта SQL, на фоне завершения разработки этого стандарта специалисты различных компаний начали работу над стандартом SQL2
- Эта работа также длилась несколько лет, было выпущено множество проектов стандарта, пока наконец в марте 1992 г. не был принят окончательный проект стандарта (SQL/92)

# Краткая история языка SQL (13)

- Этот стандарт существенно полнее стандарта SQL/89 и охватывает практически все аспекты, необходимые для реализации приложений: манипулирование схемой БД, управление транзакциями (появились точки сохранения) и сессиями (сессия – это последовательность транзакций, в пределах которой сохраняются временные отношения), подключения к БД, динамический SQL
- Наконец, были стандартизованы отношения-каталоги БД, что вообще-то не связано непосредственно с языком, но очень сильно влияет на реализацию



# Краткая история языка SQL (14)

- В 1995 г. стандарт был дополнен спецификацией интерфейса уровня вызова (Call-Level Interface – *SQL/CLI*).
  - SQL/CLI представляет собой набор спецификаций интерфейсов процедур, вызовы которых позволяют выполнять динамически задаваемые операторы SQL
  - По сути дела, SQL/CLI представляет собой альтернативу динамическому SQL
  - Интерфейсы процедур определены для всех основных языков программирования: C, Ada, Pascal, PL/1 и т. д.
- Следует заметить, что стандарт SQL/CLI послужил основой для создания повсеместно распространенных сегодня интерфейсов ODBC (Open Database Connectivity) и JDBC (Java Database Connectivity).

# Краткая история языка SQL (15)

- В 1996 г. к стандарту SQL/92 был добавлен еще один компонент – *SQL/PSM* (Persistent Stored Modules)
- Основная цель этой спецификации состоит в том, чтобы стандартизировать способы определения и использования хранимых процедур, т. е. специальным образом оформленных программ, включающих операторы SQL, которые сохраняются в базе данных, могут вызываться приложениями и выполняются внутри СУБД.

# Краткая история языка SQL (16)

- Незадолго до завершения работ по определению стандарта SQL2 была начата разработка стандарта SQL3
- Первоначально планировалось завершить проект в 1995 г. и включить в язык некоторые объектные возможности: определяемые пользователями типы данных, поддержку триггеров, поддержку темпоральных свойств данных и т. д.
- Реально работу над новым стандартом удалось частично завершить только в 1999 г., и по этой причине (а также в связи с проблемой 2000 года) стандарт получил название SQL:1999

# Краткая история языка SQL (17)

- Приведем краткую характеристику текущего состояния стандарта SQL:1999 и перспектив его развития
- Прежде всего, заметим, что каждый новый вариант стандарта языка SQL был существенно объемнее предыдущих версий. Так, если стандарт SQL/89 занимал около 600 страниц, то объем SQL/92 составлял на 300 с лишним страниц больше
- Самые первые проекты SQL<sub>3</sub> занимали около 1500 страниц

# Краткая история языка SQL (18)

- Это вполне естественно, потому что язык усложняется, а его спецификации становятся более детальными и точными
- Но разработчики SQL3 пришли к выводу, что при таких объемах стандарта вероятность его принятия и последующей успешной поддержки заметно уменьшается
- Поэтому было принято решение разбить стандарт на относительно независимые части, которые можно было бы разрабатывать и поддерживать по отдельности

# Краткая история языка SQL (16)

- В 1999 г. были приняты пять первых частей стандарта SQL:1999
- Первая часть (SQL/Framework) посвящена описанию концептуальной структуры стандарта
- В этой части приводится развернутая аннотация следующих четырех частей и формулируются требования к реализациям, претендующим на соответствие стандарту

# Краткая история языка SQL (19)

- Вторая часть SQL:1999 (SQL/Foundation) образует базис стандарта. Вводится система типов языка, формулируются правила определения функциональных зависимостей и возможных ключей, определяются синтаксис и семантика основных операторов SQL:
  - операторов определения и манипулирования схемой базы данных;
  - операторов манипулирования данными;
  - операторов управления транзакциями;
  - операторов управления подключениями к базе данных и т. д.

# Краткая история языка SQL (20)

- Третью часть занимает уточненная по сравнению с SQL/92 спецификация SQL/CLI
- В четвертой части специфицируется SQL/PSM – синтаксис и семантика языка определения хранимых процедур
- Наконец, в пятой части – SQL/Bindings – определяются правила связывания SQL для стандартных версий языков программирования FORTRAN, COBOL, PL/1, Pascal, Ada, C и MUMPS



# Краткая история языка SQL (21)

- В стандарт SQL:1999 должны были войти еще несколько частей. Среди них спецификации следующих средств:
  - управление распределенными транзакциями (SQL/Transaction);
  - поддержка темпоральных свойств данных (SQL/Temporal);
  - управление внешними данными (SQL/MED);
  - связывание с объектно-ориентированными языками программирования (SQL/OLB);
  - поддержка оперативной аналитической обработки (SQL/OLAP)

# Краткая история языка SQL (22)

- В конце 2003 г. был принят и опубликован новый вариант международного стандарта *SQL:2003*
- Многие специалисты считали, что в варианте стандарта, следующем за *SQL:1999*, будут всего лишь исправлены неточности *SQL:1999*
- Но на самом деле, в *SQL:2003* специфицирован ряд новых и важных свойств, часть из которых мы затронем в этом курсе

# Краткая история языка SQL (23)

- Претерпела некоторые изменения общая организация стандарта
- Стандарт SQL:2003 состоит из следующих частей:
  - 9075-1, SQL/Framework;
  - 9075-2, SQL/Foundation;
  - 9075-3, SQL/CLI;
  - 9075-4, SQL/PSM;
  - 9075-9, SQL/MED;
  - 9075-10, SQL/OLB;
  - 9075-11, SQL/Schemata;
  - 9075-13, SQL/JRT;
  - 9075-14, SQL/XML

# Краткая история языка SQL (24)

- Части 1-4 и 9-10 с необходимыми изменениями остались такими же, как и в SQL:1999
- Часть 5 (SQL/Bindings) перестала существовать; соответствующие спецификации включены в часть 2
- Раздел части 2 SQL:1999, посвященный информационной схеме, выделен в отдельную часть 11
- Появились две новые части – 13 и 14

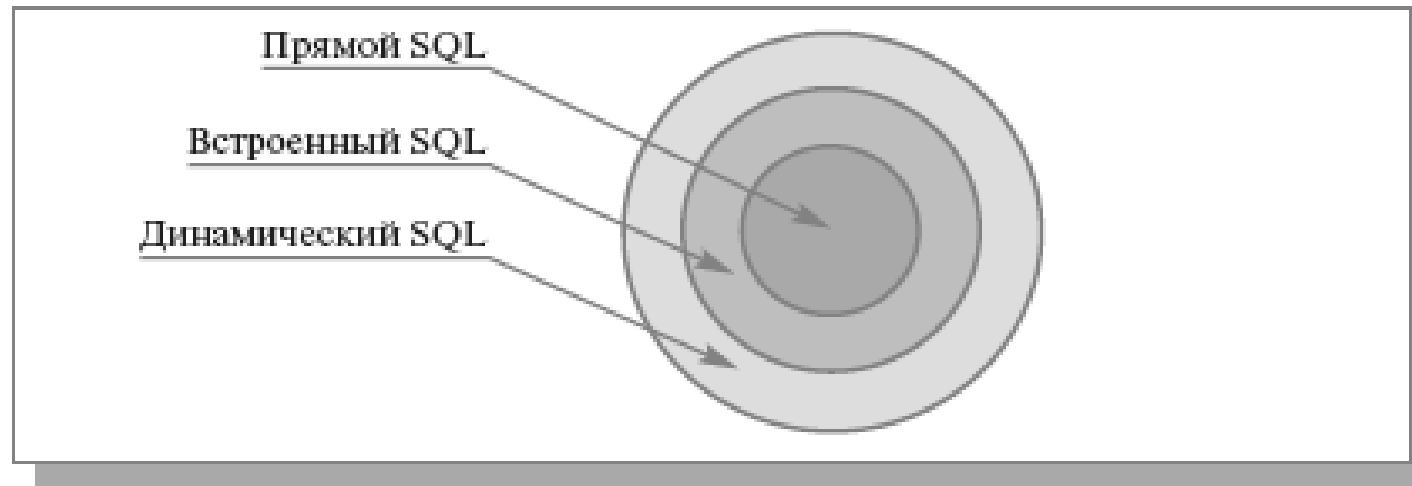
# Краткая история языка SQL (25)

- Часть 13 полностью называется «SQL Routines and Types Using the Java Programming Language» («Использование подпрограмм и типов SQL в языке программирования Java»)
  - появление такой части стандарта оправдано повышенным вниманием к языку Java со стороны ведущих производителей SQL-ориентированных СУБД
- Наконец, последняя часть SQL:2003 посвящена спецификациям языковых средств, позволяющих работать с XML-документами в среде SQL

# Краткая история языка SQL (26)

- Текущее состояние процесса стандартизации языка SQL отражает текущее состояние технологии SQL-ориентированных баз данных
- Ведущие поставщики соответствующих СУБД (сегодня это компании IBM, Oracle и Microsoft) стараются максимально быстро реагировать на потребности и конъюнктуру рынка и расширяют свои продукты все новыми и новыми возможностями
- Очевидна потребность в стандартизации соответствующих языковых средств, но процесс стандартизации явно не поспевает за происходящими изменениями

# Структура языка SQL (1)



- В данной лекции мы начинаем систематически описывать базовые механизмы языка SQL
- Чтобы пояснить, о какой части языка пойдет речь в этой и следующих лекциях, обратимся к рисунку

# Структура языка SQL (2)

- Язык SQL, соответствующий последним стандартам SQL:2003, SQL:1999 (и даже SQL/92), это очень богатый и сложный язык, все возможности которого трудно сразу осознать и тем более понять
- Поэтому приходится разбивать язык на уровни, или слои, такие, что каждый уровень языка включает все конструкции, входящие в более низкие уровни
- В стандарте определяется несколько способов разбиения языка на уровни
- В одной из классификаций язык разбивается на *базовый (entry)*, *промежуточный (intermediate)* и *полный (full)* уровни



# Структура языка SQL (4)

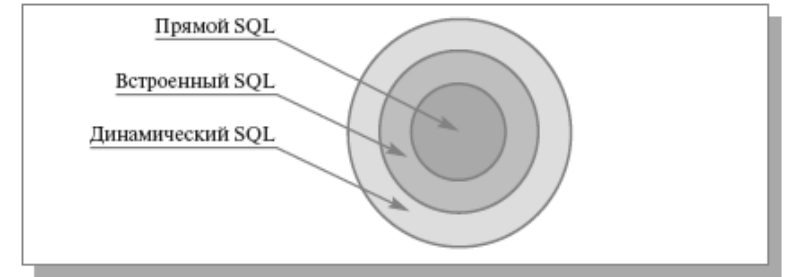
- Эта классификация ориентирована, прежде всего, на производителей СУБД, в которых поддерживается SQL
- Реализация базового уровня языка является обязательным условием хотя бы какого-то соответствия стандарту
- Реализация промежуточного уровня желательна, и обычно именно такой уровень языка поддерживается ведущими компаниями-производителями SQL-ориентированных СУБД
- Наконец, полный уровень языка является целью, к достижению которой следует стремиться

# Структура языка SQL (5)

- Критерием отнесения той или иной возможности языка к некоторому уровню является оцениваемая создателями стандарта SQL (большая часть которых является сотрудниками ведущих компаний, производящих SQL-ориентированные СУБД) техническая сложность реализации этой возможности
- Конечно, такая классификация важна и для программистов приложений баз данных, но только для того, чтобы оценить реальные возможности конкретной СУБД
- Для понимания языка SQL это разбиение на уровни несущественно

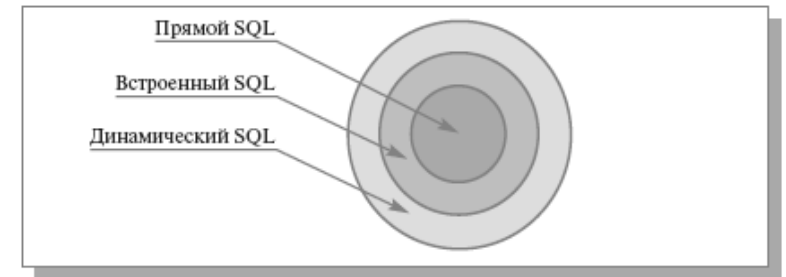
# Структура языка SQL (6)

- Другая классификация показана на рисунке
- Среди всех конструкций языка SQL можно выделить такие конструкции, которые можно было использовать при *прямом (direct)* взаимодействии конечного пользователя с СУБД (например, в интерактивном режиме)
- В некотором смысле этот уровень также является базовым, поскольку соответствующие средства языка в наибольшей степени отражают его ориентированность на работу с мультимножествами



# Структура языка SQL (7)

- На следующем уровне, уровне встраиваемого (embedded) SQL, язык расширяется конструкциями, позволяющими использовать возможности прямого SQL в программах, написанных на традиционных языках программирования
- Наконец, на уровне динамического (dynamic) SQL во встраиваемый SQL добавляются конструкции, позволяющие приложениям обращаться к СУБД с конструкциями прямого SQL, которые динамически образуются во время выполнения программ



# Структура языка SQL (8)

- Вторая классификация является более полезной для читателя, постигающего основы языка SQL
- Дополнительные возможности, присутствующие во встраиваемом и в динамическом SQL, не слишком сильно влияют на модельное представление языка
- Конечно, возможности встраиваемого и динамического SQL необходимо хорошо знать разработчикам приложений SQL-ориентированных баз данных
- Но поскольку задачей этого курса не является обучение использованию языка SQL при программировании приложений баз данных, мы не будем затрагивать эти темы
- Обратимся к прямому SQL, причем не в полном объеме стандартов SQL:2003 и SQL:1999 (этого не позволяет сделать объем курса)
- Обсудим только наиболее важные аспекты