

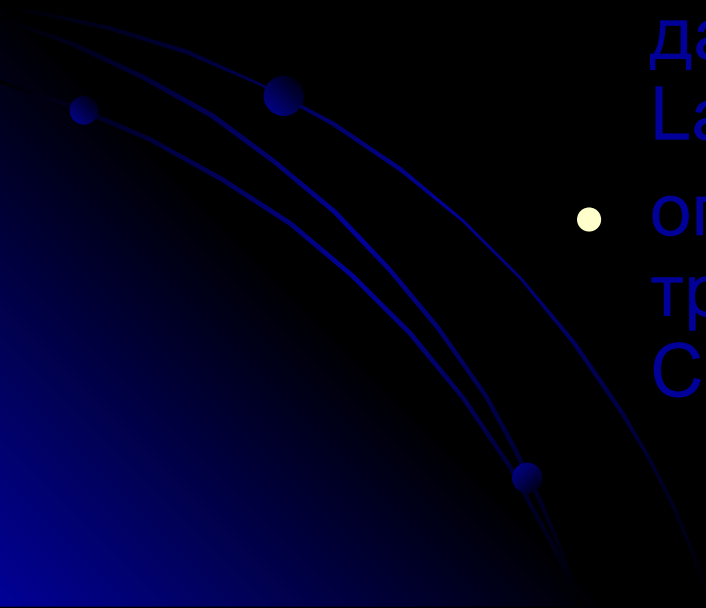


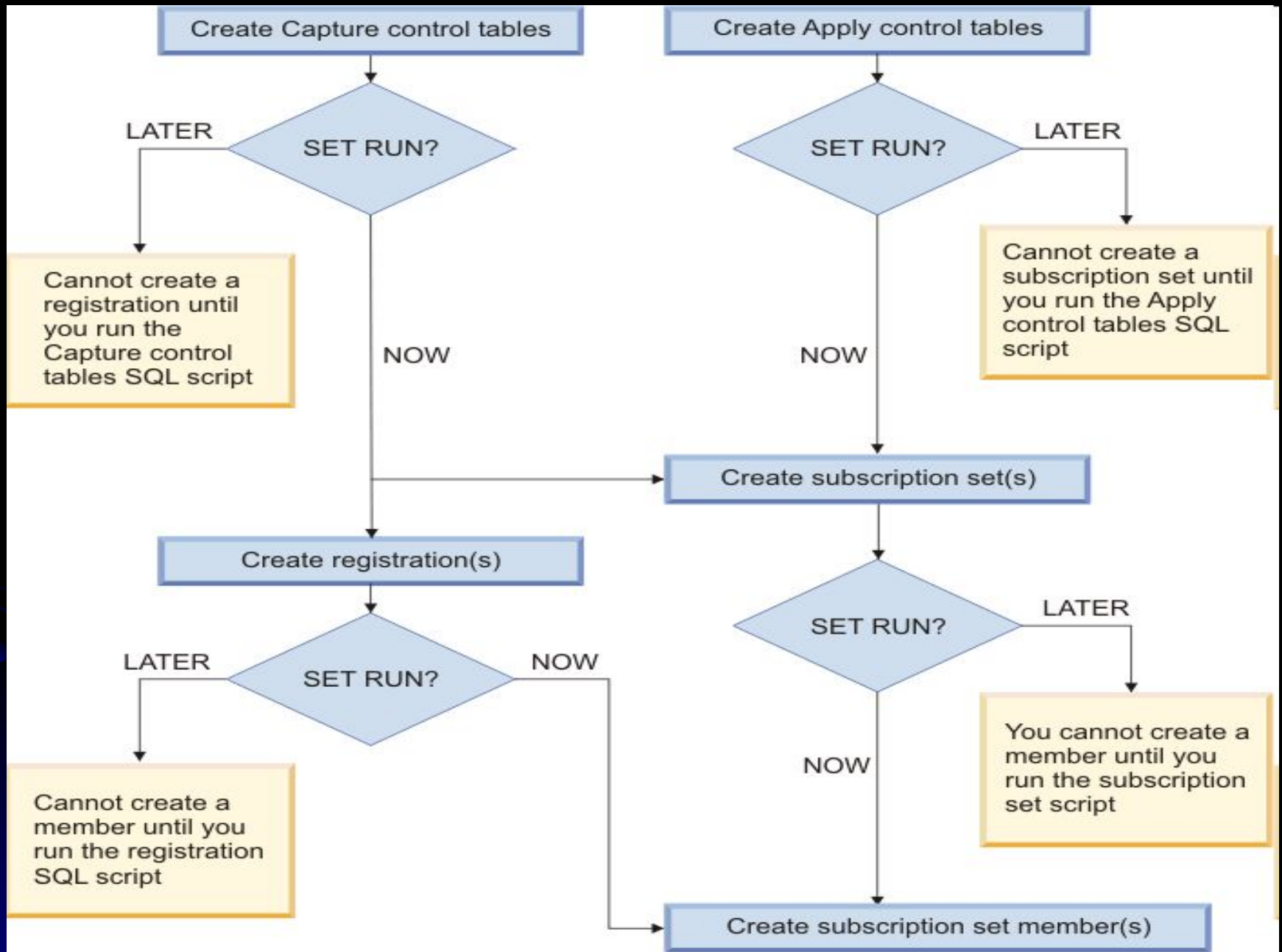
SQL (англ. Structured Query Language — язык структурированных запросов) — универсальный компьютерный язык, применяемый для создания, модификации и управления данными в реляционных базах данных. Вопреки существующим заблуждениям, SQL является информационно-логическим языком, а не языком программирования.



Язык SQL делится на четыре части:

- операторы определения данных (англ. Data Definition Language, DDL)
- операторы манипуляции данными (англ. Data Manipulation Language, DML)
- операторы определения доступа к данным (англ. Data Control Language, DCL)
- операторы управления транзакциями (англ. Transaction Control Language, TCL)





# Немного истории

В начале 1970-х годов в одной из исследовательских лабораторий компании IBM была разработана экспериментальная реляционная СУБД System R (англ.), для которой затем был создан специальный язык SEQUEL, позволявший относительно просто управлять данными в этой СУБД. Аббревиатура SEQUEL расшифровывалась как англ. Structured English QUery Language — «структурированный английский язык запросов». Позже по юридическим соображениям язык SEQUEL был переименован в SQL. Когда в 1986 году первый стандарт языка SQL был принят ANSI (American National Standards Institute), официальным произношением стало [ˌes kjuːˈel] — эс-кью-эл. Несмотря на это, даже англоязычные специалисты по прежнему часто называют SQL сиквел, вместо эс-кью-эл (по-русски также часто говорят «эс-ку-эль»).

Целью разработки было создание простого непроцедурного языка, которым мог воспользоваться любой пользователь, даже не имеющий навыков программирования. Собственно разработкой языка запросов занимались Дональд Чэмбэрлин (Donald D. Chamberlin) и Рэй Бойс (Ray Boyce). Пэт Селинджер (Pat Selinger) занималась разработкой стоимостного оптимизатора (англ. cost-based optimizer), Рэймонд Лори (Raymond Lorie) занимался компилятором запросов.

Стоит отметить, что SEQUEL был не единственным языком подобного назначения. В Калифорнийском Университете Беркли была разработана некоммерческая СУБД Ingres (являвшаяся, между прочим, дальним прародителем популярной сейчас некоммерческой СУБД PostgreSQL), которая являлась реляционной СУБД, но использовала свой собственный язык QUEL, который, однако, не выдержал конкуренции по количеству поддерживающих его СУБД с языком SQL.

# Преимущества

- **Независимость от конкретной СУБД:** Несмотря на наличие диалектов и различий в синтаксисе, в большинстве своём тексты SQL-запросов, содержащие DDL и DML, могут быть достаточно легко перенесены из одной СУБД в другую. Существуют системы, разработчики которых изначально закладывались на применение по меньшей мере нескольких СУБД (например: система электронного документооборота Documentum может работать как с Oracle, так и с Microsoft SQL Server и IBM DB2). Естественно, что при применении некоторых специфичных для реализации возможностей такой переносимости добиться уже очень трудно.
- **Наличие стандартов:** Наличие стандартов и набора тестов для выявления совместимости и соответствия конкретной реализации SQL общепринятому стандарту только способствует «стабилизации» языка. Правда, стоит обратить внимание, что сам по себе стандарт местами чересчур формализован и раздут в размерах, например, Core-часть стандарта SQL:2003 представляет собой более 1300 страниц текста.
- **Декларативность:** С помощью SQL программист описывает только то, какие данные нужно извлечь или модифицировать. То, каким образом это сделать, решает СУБД непосредственно при обработке SQL-запроса. Однако не стоит думать, что это полностью универсальный принцип — программист описывает набор данных для выборки или модификации, однако ему при этом полезно представлять, как СУБД будет разбирать текст его запроса. Особенно критичны такие моменты становятся при работе с большими базами данных и со сложными запросами — чем сложнее сконструирован запрос, тем больше он допускает вариантов написания, различных по скорости выполнения, но одинаковых по итоговому набору данных.

# Недостатки

Несоответствие реляционной модели данных:  
Создатель реляционной модели данных Эдгар Кодд, Кристофер Дейт и их сторонники указывают на то, что SQL не является истинно реляционным языком. В частности они указывают на следующие проблемы SQL[3]:

- Повторяющиеся строки
- Неопределённые значения (nulls)
- Явное указание порядка колонок слева направо
- Колонки без имени и дублирующиеся имена колонок
- Отсутствие поддержки свойства «=»
- Использование указателей
- Высокая избыточность

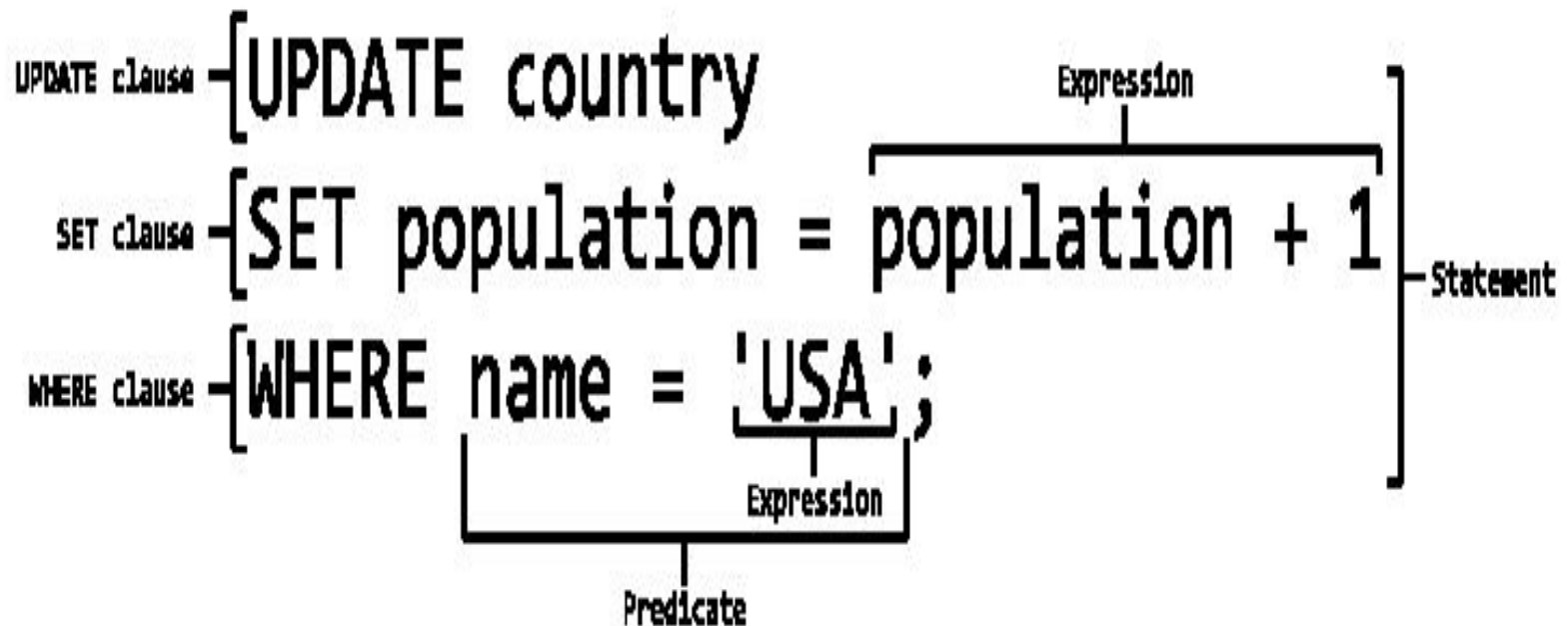
# Элементы языка

В SQL язык подразделяется на несколько языковых элементов, в том числе:

- Положения, которые в некоторых случаях необязательно, составные компоненты заявлений и запросов. [16]
- Выражения, которые могут привести либо скалярные значения или таблицы, состоящей из столбцов и строк данных.
- Предикатов которые определяют условия, которые могут быть оценены по SQL трех-значной логике (3VL) булева истина и ценностей, которые используются для ограничения последствий заявления и запросы, или изменить программу потока.
- Запросы, которые Получить данные, на основе определенных критериев.
- Заявления, которые могут иметь стойкий эффект от схем и данных, или которые могут контролировать операции, программа потоков соединений, сессий, или диагностика.
- SQL заявления также включать запятой (";") заявление терминаторе. Хотя это и не требуется на любой платформе, он определяется как стандартный рамках SQL грамматики.
- Незначительные пробелы обычно игнорируется в SQL заявления и запросов, что упрощает для форматирования кода SQL для удобочитаемости.



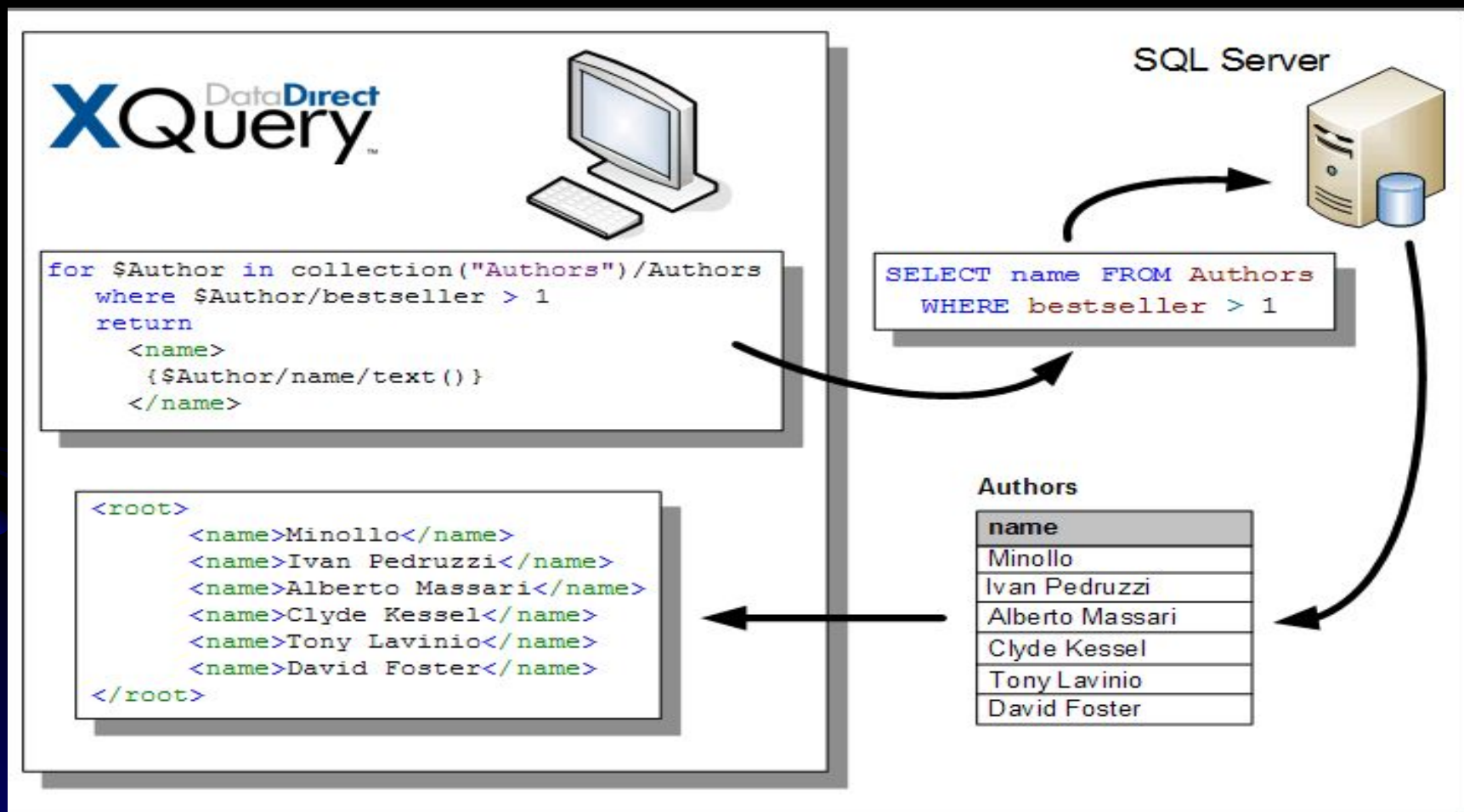
Эта диаграмма показывает некоторые из элементов языка SQL, которые составляют единое заявление.



# Альтернативы SQL

- Следует проводить различие между альтернативами для реляционных языков запросов и альтернатив SQL. Ниже предлагаются альтернативы SQL, но до сих пор (номинально) реляционный.
- QL - объектно-ориентированный Datalog
- Язык запросов 4D (4D QL)
- Aldat Реляционная алгебра и домен алгебра
- Datalog
- Язык запросов Hibernate (HQL) - A Java-инструмент, который использует измененные SQL
- IBM Business System 12 (IBM BS12)
- ISBL
- Java Стойкость Язык запросов (JPQL) - язык запросов, используемые в Java API Стойкость в Java EE5
- LINQ
- Объект Язык запросов
- QBE (запрос по образцу), созданный Моше Zloof, IBM 1977
- QLC - запрос интерфейса для Mnesia, CTB, Дек и т.д. (Erlang язык программирования)
- QUEL представил в 1974 году UC Berkeley Энгр проекта.
- Учебник D
- XQuery

# SQL предназначен для работ с реляционными базами данных.



THANK YOU  
FOR YOUR  
ATTENTION!

