## Учреждение образования

## Белорусский государственный технологический университет

Кафедра полиграфического оборудования и

системы обработки информации

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №9**

по дисциплине «Стандартизация и сертификация

информационных систем и технологий»

Тема

«Ознакомление со стандартом *SQL*»

Выполнил студент:

Трошко Валерия

ф-та ИТ 4 к. 5 гр. ПОИТ

Проверил

кандидат технических наук

Сулим П.Е.

Отчет по лабораторной работе

защищен с отметкой баллов

Минск 2023

1. **Стандарт *SQL*.**

***SQL*** (*structured query language*  – «язык структурированных запросов») – [декларативный](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BA%D0%BB%D0%B0%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), применяемый для создания, модификации и управления данными в [реляционной базе данных](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), управляемой соответствующей [системой управления базами данных](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85).

Является, прежде всего, [информационно-логическим языком](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA), предназначенным для описания, изменения и извлечения данных, хранимых в [реляционных базах данных](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85). Изначально *SQL* был основным способом работы пользователя с [базой данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) и позволял выполнять следующий набор операций:

* создание в базе данных новой таблицы;
* добавление в таблицу новых записей;
* изменение записей;
* удаление записей;
* выборка записей из одной или нескольких таблиц;
* изменение структур таблиц.

Со временем *SQL* усложнился – обогатился новыми конструкциями, обеспечил возможность описания и управления новыми хранимыми объектами (например, индексы, представления, триггеры и хранимые процедуры) – и стал приобретать черты, свойственные языкам программирования.

При всех своих изменениях *SQL* остаётся самым распространённым лингвистическим средством для взаимодействия прикладного программного обеспечения с базами данных. В то же время современные СУБД, а также информационные системы, использующие СУБД, предоставляют пользователю развитые средства визуального построения запросов.

Поскольку к началу 1980-х годов существовало несколько вариантов СУБД от разных производителей, причём каждый из них обладал собственной реализацией языка запросов, было принято решение разработать стандарт языка, который будет гарантировать переносимость ПО с одной СУБД на другую (при условии, что они будут поддерживать этот стандарт).

В 1983 году Международная организация по стандартизации (*ISO*) и Американский национальный институт стандартов (*ANSI*) приступили к разработке стандарта языка *SQL*. По прошествии множества консультаций и отклонения нескольких предварительных вариантов, в 1986 году *ANSI* представил свою первую версию стандарта, описанную в документе *ANSI* *X3*.*135-1986* под названием «*Database Language SQL*» (язык баз данных *SQL*). Неофициально этот стандарт *SQL-86* получил название *SQL1*. Год спустя была завершена работа над версией стандарта *ISO* *9075-1987* под тем же названием. Разработка этого стандарта велась под патронажем Технического Комитета *TC97* (англ. *Technical Committee TC97*), областью деятельности которого являлись процессы вычисления и обработки информации (англ. Computing and Information Processing). Именно его подразделение, именуемое как Подкомитет *SC21* (англ. Subcommittee *SC21*), курировало разработку стандарта, что стало залогом идентичности стандартов *ISO* и *ANSI* для *SQL1* (*SQL-86*).

Стандарт *SQL1* разделялся на два уровня. Первый уровень представлял собой подмножество второго уровня, описывавшего весь документ в целом. То есть, такая структура предусматривала, что не все спецификации стандарта *SQL1* будут относиться к Уровню 1. Тем самым поставщик, заявлявший о поддержке данного стандарта, должен был заявлять об уровне, которому соответствует его реализация языка *SQL*. Это значительно облегчило принятие и поддержку стандарта, поскольку производители могли реализовывать его поддержку в два этапа.

Со временем к стандарту накопилось несколько замечаний и пожеланий, особенно с точки зрения обеспечения целостности и корректности данных, в результате чего в 1989 году данный стандарт был расширен, получив название *SQL89*. В частности, в него была добавлена концепция первичного и внешнего ключей. *ISO*-версия документа получила название *ISO* 9075:1989 «*Database Language SQL with Integrity Enhancements*» (Язык баз данных *SQL* с добавлением контроля целостности). Параллельно была закончена и *ANSI*-версия.

Сразу после завершения работы над стандартом *SQL1* в 1987 году была начата работа над новой версией стандарта, который должен был заменить стандарт *SQL89*, получив название *SQL2*, поскольку дата принятия документа на тот момент была неизвестна. Таким образом, фактически *SQL89* и *SQL2* разрабатывались параллельно. Новая версия стандарта была принята в 1992 году, заменив стандарт SQL89. Новый стандарт, озаглавленный как *SQL92*, представлял собой по сути расширение стандарта *SQL1*, включив в себя множество дополнений, имевшихся в предыдущих версиях инструкций.

Как и *SQL1*, *SQL92* также был разделён на несколько уровней, однако, во-первых, число уровней было увеличено с двух до трёх, а во-вторых, они получили названия вместо порядковых цифр: начальный (англ. *entry*), средний (англ. *intermediate*), полный (англ. *full*). Уровень «полный», как и Уровень 2 в *SQL1* подразумевал весь стандарт целиком. Уровень «начальный» представлял собой подмножество уровня «средний», в свою очередь, представлявшего собой подмножество уровня «полный». Уровень «начальный» был сравним с Уровнем 2 стандарта *SQL1*, но спецификации этого уровня были несколько расширены. Таким образом, цепочка включений уровней стандартов выглядела примерно следующим образом: *SQL1* Уровень 1 → *SQL1* Уровень 2 → *SQL92* «Начальный» → *SQL92* «Средний» → *SQL92* «Полный».

После принятия стандарта *SQL92* к нему были добавлены ещё несколько документов, расширявших функциональность языка. Так, в 1995 году был принят стандарт *SQL/CLI* (*Call Level Interface*, интерфейс уровня вызовов), впоследствии переименованный в CLI95. На следующий год был принят стандарт *SQL/PSM* (*Persistent Stored Modules*, постоянно хранимые модули), получивший название *PSM-96*.

Следующим стандартом стал *SQL:1999* (*SQL3)*. В настоящее время действует стандарт, принятый в 2003 году (*SQL:2003*) с небольшими модификациями, внесёнными позже (*SQL:2008*).

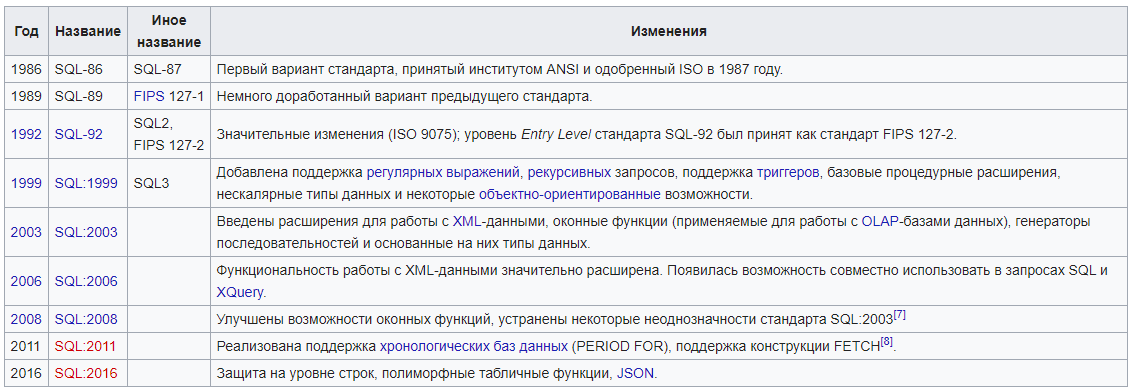


Рисунок 1 – История версий стандарта SQL

Язык SQL представляет собой совокупность операторов, инструкций, вычисляемых функций. Согласно общепринятому [стилю программирования](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B8%D0%BB%D1%8C_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), операторы (и другие зарезервированные слова) в *SQL* обычно рекомендуется писать [прописными буквами](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%B3%D0%BB%D0%B0%D0%B2%D0%BD%D1%8B%D0%B5_%D0%B1%D1%83%D0%BA%D0%B2%D1%8B).

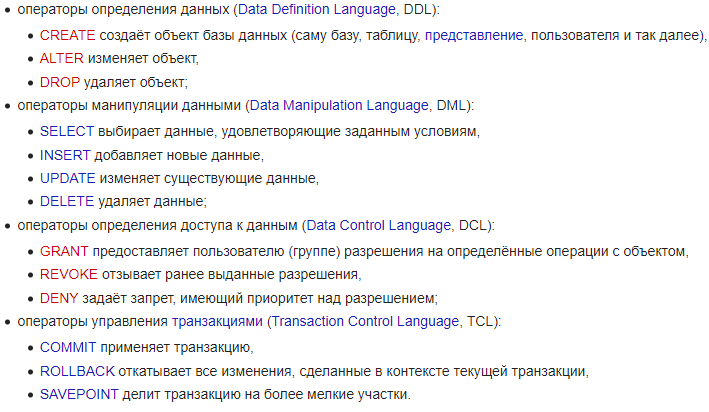


Рисунок 2 – Операторы SQL

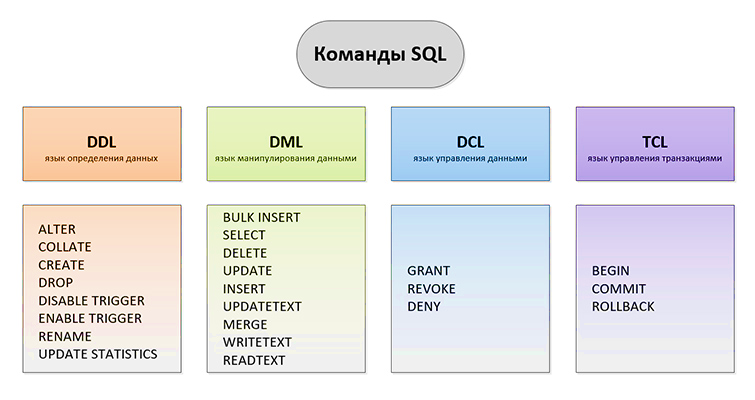


Рисунок 3 – Команды SQL

Несмотря на наличие международного стандарта ANSI SQL, многие компании, занимающиеся разработкой СУБД, вносят изменения в язык *SQL*. Каждая из реализаций языка *SQL* в конкретной СУБД называется *диалектом* (некоторое подмножество стандарта *SQL*). Функции, которые добавляются к стандарту языка разработчиками коммерческих реализаций – это расширения. В стандарте языка *SQL* определены конкретные типы данных, которые могут хранится в БД. Во многих реализациях этот список расширяется за счет дополнений. 3 уровня соответствия стандарту *ANSI/ISO* – начальный, промежуточный и полный.

Производители СУБД (*Oracle, Microsoft, Borland, Informix, Sybase*) применяют реализации *SQL*, отвечающие как минимум начальному уровню и содержащие некоторые расширения, специфические для данной СУБД. Не существует двух совершенно идентичных диалектов. И поскольку разработчики вводят в системы все новые средства, они расширяют свои диалекты языка *SQL*, в результате чего отдельные диалекты больше отличаются друг от друга. Достоинства: более широкие возможности, больше типов данных и количества команд, больше дополнительных возможностей у имеющихся команд. Кроме того, такие нестандартные возможности языка со временем могут быть вклучены в стандарт. Недостаток в том, что различия в синтаксисе реализаций *SQL* затрудняют перенос приложений из одной системы в другую. В широко распространенных в настоящее время СУБД используются следующие диалекты языка *SQL*: *PL/SQL* – в СУБД *Oracle*; *Transact-SQL* – в СУБД *Microsoft SQL*; *Informix-SQL* – в СУБД *Informix*; *Jet SQL* – *Microsoft Access*. Язык *Jet SQL* почти соответствует стандарту ANSI SQL. Основные различия языков *Jet SQL* и *ANSI SQL* состоят в следующем:

* они имеют разные наборы зарезервированных слов и типов данных;
* разные правила применимы к оператору *Between*, используемому для определения условий выборки записей;
* подстановочные знаки *ANSI* и *Microsoft Jet*, которые используются в операторе *Like*, различны;
* язык *Jet SQL* обычно предоставляет пользователю большую свободу, например, разрешается группировка и сортировка по значению выражения; язык *Jet SQL* позволяет использовать более сложные выражения.

1. **Диалекты *SQL. DSL*.**

Предметно-ориентированные языки (***DSL***) – это языки программирования с более высоким уровнем абстракции, которые отражают специфику решаемых с их помощью задач. Такие языки оперируют понятиями и правилами из определенной предметной области.

Чем предметно-ориентированные языки отличаются от «настоящих» языков программирования? Предметно-ориентированные языки обычно не такие сложные, как языки общего назначения, например, *Java*, *C* или *Ruby*. Как правило, предметно-ориентированные языки разрабатываются в сотрудничестве со специалистами в той сфере деятельности, для которой проектируется язык. Зачастую для использования таких языков не требуется квалификация разработчика и на них программируют люди, хорошо разбирающиеся в предметной области.

В чем преимущество предметно-ориентированных языков? Зачем они нужны? Использование *DSL* несет в себе целый ряд преимуществ. Самое очевидное из них заключается в том, что использование языка и движка трансформации позволит вам значительно повысить **эффективность** определенного этапа процесса разработки, поскольку вам больше не придется вручную выполнять утомительную работу.

В отличие от интерпретации, генерация исходного кода из программ, написанных на *DSL*, позволяет использовать специализированные абстракции **без каких-либо накладных расходов при выполнении кода**, поскольку генератор, подобно компилятору, способен удалять абстракции и генерировать высокопроизводительный код. Благодаря тому, что вы сможете выражать понятия из нужной вам области на языке, который тесно с ней связан, **продумывать логику станет легче**, поскольку код будет свободен от подробностей реализации. Иными словами, использование *DSL* позволит вам отделить главное от второстепенных деталей.

Благодаря тому, что нотации, набор абстракций и область применения предметно-ориентированных языков максимально соответствуют соглашениям, принятым среди экспертов в соответствующей области   
(т.е. непрограммистов), такие языки способствуют тесной **интеграции между разработчиками и специалистами других направленностей**.

Использование предметно-ориентированных языков в совокупности с механизмом выполнения позволяет сделать логику приложения, написанного на *DSL*-коде, **независимой от целевой платформы**. Использование *DSL* также позволяет повысить **качество** создаваемого продукта: снизить число ошибок, обеспечить более точное соответствие архитектуры техническому заданию и повысить легкость сопровождения. Это достигается благодаря упразднению «лишних» степеней свободы и дублирующегося кода, а также автоматизации однообразной работы.

|  |  |
| --- | --- |
|  | Картинки по запросу "sql язык код" |
| Рисунок 4 – Пример DSL | Рисунок 5 – Пример SQL |

**Информация с сайта:** <https://ru.wikipedia.org/wiki/SQL>

**Вывод:**

***SQL*** (*structured query language*  – «язык структурированных запросов») – [декларативный](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%BA%D0%BB%D0%B0%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) [язык программирования](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F), применяемый для создания, модификации и управления данными в [реляционной базе данных](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), управляемой соответствующей [системой управления базами данных](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85).

Является, прежде всего, [информационно-логическим языком](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA), предназначенным для описания, изменения и извлечения данных, хранимых в [реляционных базах данных](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85). Изначально *SQL* был основным способом работы пользователя с [базой данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) и позволял выполнять следующий набор операций: создание в базе данных новой таблицы, добавление в таблицу новых записей, изменение записей, удаление записей, выборка записей из одной или нескольких таблиц, изменение структур таблиц.

В 1983 году Международная организация по стандартизации (*ISO*) и Американский национальный институт стандартов (*ANSI*) приступили к разработке стандарта языка *SQL*.

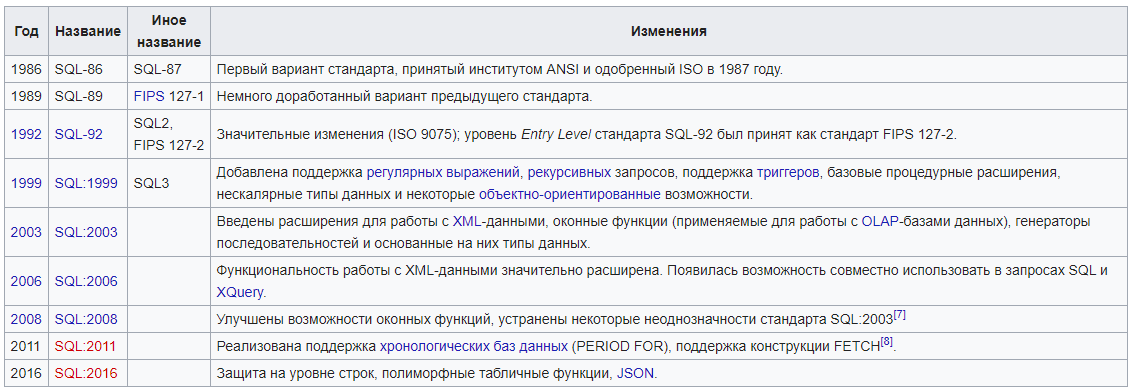


Рисунок 6 – История версий стандарта SQL

Несмотря на наличие международного стандарта ANSI SQL, многие компании, занимающиеся разработкой СУБД, вносят изменения в язык *SQL*. Каждая из реализаций языка *SQL* в конкретной СУБД называется *диалектом* (некоторое подмножество стандарта *SQL*).

Предметно-ориентированные языки (***DSL***) – это языки программирования с более высоким уровнем абстракции, которые отражают специфику решаемых с их помощью задач. Такие языки оперируют понятиями и правилами из определенной предметной области.

Преимущества предметно-ориентированных языков:

* повышают эффективность;
* используются без каких-либо расходов при выполнении кода;
* продумывать логику стало легче;
* способствуют тесной интеграции между разработчиками и специалистами других направленностей;
* независимы от целевой платформы;
* позволяют повысить качество создаваемого продукта: снизить число ошибок, обеспечить более точное соответствие архитектуры техническому заданию и повысить легкость сопровождения.