СТРУКТУРА ЯЗЫКА SQL

**Введение**

Сегодня во многих организациях основной упор делается на построение автоматизированных информационных систем (АИС) обеспечивающих наличие точной и самой свежей информации. Без АИС в большинстве организаций невозможно выполнять обычные, повседневные операции и создавать достаточное количество итоговых отчетов, помогающих руководству принимать стратегические решения. Для того чтобы информация АИС была полезна, она должна быть точна, полна, и организована так, чтобы данные можно было считывать при необходимости и в нужном формате.

Основой своевременных АИС являются реляционные базы данных (БД). Одним из основных преимуществ реляционного подхода к организации баз данных является то, что пользователи реляционных БД получают возможность эффективной работы в терминах простых и наглядных понятий таблиц, их строк и столбцов без потребности знания реальной организации данных во внешней памяти.

Реляционная модель данных, содержащая набор четких предписаний к базовой организации любой реляционной системы управления базами данных (СУБД), позволяет пользователям работать в ненавигационной манере, т.е. для выборки информации из БД человек должен всего лишь указать список интересующих его таблиц и те условия, которым должны удовлетворять выбираемые данные. СУБД скрывает от пользователя выполняемые ей последовательные просмотры таблиц, выполняя их наиболее эффективным образом. Очень важная особенность реляционных систем состоит в том, что результатом выполнения любого запроса к таблицам БД является также таблица, которую можно сохранить в БД и / или по отношению к которой можно выполнять новые запросы.

Базовым требованием к реляционным СУБД является наличие мощного и в тоже время простого языка, позволяющего выполнять все необходимые пользователям операции. В последние годы таким повсеместно принятым языком стал язык реляционных БД SQL - Structured Query Language.

Structured Query Language представляет собой непроцедурный язык, используемый для управления данными реляционных СУБД. Термин «непроцедурный» означает, что на данном языке можно сформулировать, что нужно сделать с данными, но нельзя проинструктировать, как именно это следует сделать. Иными словами, в этом языке отсутствуют алгоритмические конструкции, такие как метки, операторы цикла, условные переходы и др.

Команды в SQL могут работать со всеми группами таблиц как с единым объектом и могут обрабатывать любое количество информации, извлеченной или полученной из их, в виде единого модуля.

До появления SQL в СУБД (независимо от того, на какой модели они основывались) приходилось поддерживать по крайней мере три языка, которые обычно имели мало общего: язык определения данных (ЯОД), служащий для спецификации структур БД (обычно общую структуру БД называют схемой БД); язык манипулирования данными (ЯМД), позволяющий создавать прикладные программы, взаимодействующие с БД; и язык администрирования БД (ЯАДБ), с помощью которого можно было выполнять служебные действия (например, изменять структуру БД или производить ее настройку с целью повышения эффективности). Кроме того, если требовалось предоставить пользователям СУБД интерактивный доступ к БД, приходилось вводить еще один язык, операторы которого выполняются в диалоговом режиме. Язык SQL позволяет решать все эти задачи.

Следует отметить, что к достоинствам языка SQL относится наличие международных стандартов. Первый международный стандарт был принят в 1989 г., и соответствующая версия языка называется SQL-89. Этот стандарт полностью поддерживается практически во всех современных коммерческих реляционных СУБД (например, в Informix, Sybase, Ingres, DB2 и т.д.).

Целью написания данной работы является анализ структуры языка SQL, его основных достоинств и недостатков.

**1. Общие сведения об истории создания и стандартах SQL**

Язык SQL был создан в начале 70-х годов в результате исследовательского проекта IBM, целью которого было создание языка манипуляции реляционными данными. Работа была начата сразу после появления статьи Э. Кодда в 1970 г. в лабораториях компании IBM для проверки возможностей реляционной модели. Первоначально он назывался SEQUEL (Structured English Query Language), затем - SEQUEL/2, а затем - просто SQL.

Вскоре появилась СУБД System R - экспериментальная исследовательская система с языком SEQUEL (позже SQL), созданная IBM. Затем появились многочисленные СУБД с SQL среди них:

- Oracle (Relation Software Inc. - Oracle corp.;

1981-1982 - DB2 (IBM), Ingres - CA-OpenIngres (Relation Technology Inc. - Computer Associates)

- Informix (Informix Sofrware);

- Sybase (Sybase Corp.)

Стандарт SQL определяется ANSI (Американским Национальным Институтом Стандартов) и в данное время также принимается ISO (Международной Организацией по Стандартизации). Однако, большинство коммерческих программ баз данных расширяют SQL без уведомления ANSI, добавляя разные другие особенности в этот язык, которые, как они считают, будут весьма полезны. Иногда они несколько нарушают стандарт языка, хотя хорошие идеи имеют тенденцию развиваться и вскоре становиться стандартами «рынка» сами по себе в силу полезности своих качеств.не изобретался ANSI. Это по существу изобретение IBM. Но другие компании подхватили SQL сразу же, по крайней мере одна компания (Oracle) отбила у IBM право на рыночную продажу SQL продуктов.

Первый международный стандарт был принят в 1989 г., и соответствующая версия языка называется SQL-89. Стандарт 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.

Возможно, наиболее важными достижениями стандарта SQL-89 являются четкая стандартизация синтаксиса и семантики операторов выборки и манипулирования данными и фиксация средств ограничения целостности БД, включающих возможности определения первичного и внешних ключей отношений и так называемых проверочных ограничений целостности, позволяющих сформулировать условие для каждой отдельной строки таблицы. Средства определения внешних ключей позволяют легко формулировать требования так называемой целостности БД по ссылкам. Формулировка ограничений целостности на основе понятия внешнего ключа проста и понятна.

Осознавая неполноту стандарта SQL-89, на фоне завершения разработки этого стандарта специалисты различных фирм начали работу над стандартом SQL2. Эта работа также длилась несколько лет, было выпущено несколько проектов стандарта, пока, наконец, в марте 1992 г. не был выработан окончательный проект стандарта (после чего стандарт и соответствующий язык стали называть SQL-92). Этот стандарт существенно более полный и охватывает практически все необходимые для реализации аспекты: манипулирование схемой БД, управление транзакциями и сессиями (сессия - это последовательность транзакций, в пределах которой сохраняются временные отношения), подключение к БД, динамический SQL. Наконец стандартизованы отношения-каталоги БД, что вообще-то не связано с языком непосредственно, но очень сильно влияет на реализацию. Заметим, что в стандарте представлены три уровня языка - базовый, промежуточный и полный. В течение нескольких лет после принятия стандарта производители СУБД, утверждавшие совместимость своих продуктов со стандартом, на самом деле в лучшем случае поддерживали промежуточный уровень языка SQL-92 (естественно, с собственными расширениями). Только в последних выпусках СУБД ведущих производителей обеспечивается совместимость с полным вариантом языка. Наконец, одновременно с завершением работ по определению стандарта SQL-92 была начата разработка стандарта SQL3. В конце 2003 г. был принят и опубликован новый вариант международного стандарта SQL:2003.

Ниже указан анализ основных аспектов стандартов SQL.

**2. Международный стандарт 1989 г.**

Во многих частях имеет чрезвычайно общий характер и допускает очень широкое толкование. Отсутствуют важные разделы (манипулирование схемой БД, динамический SQL, многое определяется в реализации).

Наибольшие достижения (стандартизация синтаксиса и семантики операторов выборки и манипулирования данными, фиксация средств ограничений целостности БД: определение первичного и внешнего ключей отношений, проверочные ограничения целостности)

**3. Международный стандарт 1992 г. (SQL2)**

Расширено манипулирование таблицами (Alter table, Drop table), манипулирование схемой БД (Users, Tables, Views, Columns, Domains, Table\_priveleges, Table\_constraints,).

Добавлены:

возможность управления доменами (Create domain имя char(25)… и при определении имен столбцов эти имена определяются через имена доменов);

управление транзакциями и сессиями (сессия - последовательность транзакций, в пределах которой сохраняются временные отношения);

развитие динамического SQL.

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 и послужил основой для создания повсеместно распространенных сегодня интерфейсов ODBC (Open Database Connectivity) и JDBC (Java Database Connectivity).

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.

**4. Стандарт SQL: 1999 (SQL3)**

Незадолго до завершения работ по определению стандарта SQL2 была начата разработка стандарта SQL3. 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.

г. были приняты пять первых частей стандарта SQL:1999. Первая часть (SQL/Framework) посвящена описанию концептуальной структуры стандарта: приводится развернутая аннотация следующих четырех частей.

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 (расширенные модели транзакций, контрольные точки, многозвенные транзакций);

операторов управления подключениями к базе данных и т.д.

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 (стандарт синтаксиса триггеров и 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.

В стандарт SQL: 1999 не вошли и существуют в виде отдельных стандартов:

стандарт управления распределенными транзакциями (SQL/Transaction);

стандарт поддержки темпоральных свойств данных (SQL/Temporal);

стандарт управления внешними данными (SQL/MED);

поддержка оперативной аналитической обработки (SQL/OLAP).

На рисунке 1 тезисно представлена история развития SQL.



Рисунок 1 - История развития SQL

**5. Интерактивный и вложенный SQL**

Имеются два SQL: Интерактивный и Вложенный. Большей частью, обе формы работают одинаково, но используются различно.

Интерактивный SQL используется для функционирования непосредственно в базе данных, чтобы производить вывод для использования его заказчиком. В этой форме SQL, когда вы введете команду, она сейчас же выполнится и вы сможете увидеть вывод (если он вообще получится) - немедленно.

Вложенный SQL состоит из команд SQL, помещенных внутри программ, которые обычно написаны на некотором другом языке (типа КОБОЛА или Паскаля). Это делает эти программы более мощными и эффективным. Однако, допуская эти языки, приходится иметь дело со структурой SQL и стилем управления данных который требует некоторых расширений к интерактивному SQL. Передача SQL команд во вложенный SQL является выдаваемой («passed off») для переменных или параметров используемых программой в которую они были вложены.

Чтобы вложить SQL в другой язык, нужно использовать пакет программ, который бы обеспечивал поддержку вложения SQL в этот языки конечно же, поддержку самого языка. Главным образом, е используются команды SQL для работы в таблицах базы данных, передачи результатов вывода в программу и получения ввода из программы в которую они вкладываются, обобщенно ссылаясь к главной программе (которая может или не может принимать их из диалога или посылать обратно в диалог пользователя и программы).

**6. Cубподразделения SQL**

И в интерактивной, и во вложенной формах SQL, имеются многочисленные части, или субподразделения. К сожалению, эти термины не используются повсеместно во всех реализациях. Они подчеркиваются ANSI и полезны на концептуальном уровне, но большинство SQL программ практически не обрабатывающих отдельно, так что они по существу становятся функциональными категориями команд SQL.(Язык Определения Данных) - так называемый Язык Описания Схемы в ANSI, состоит из команд, которые создают объекты (таблицы, индексы, просмотры, итак далее) в базе данных.(Язык Манипулирования Данными) - это набор команд, которые определяют, какие значения представлены в таблицах в любой момент времени.(Язык Управления Данными) состоит из средств, которые определяют, разрешить ли пользователю выполнять определенные действия или нет.

Данное разделение является лишь примерным, не существует точного канонического разделения.

**7. DML**

SQL символизирует собой Структурированный Язык Запросов. Запросы - вероятно, наиболее часто используемый аспект SQL. Фактически, для категории SQL пользователей, маловероятно чтобы кто-либо использовал этот язык для чего-то другого.

Запрос - команда, которую вы даете вашей программе базы данных, и которая сообщает ей, чтобы она вывела определенную информацию из таблиц в память. Эта информация обычно посылается непосредственно на экран компьютера или терминала, которым вы пользуетесь, хотя, в большинстве случаев, ее можно также послать принтеру, сохранить в файле (как объект в памяти компьютера), или представить как вводную информацию для другой команды или процесса.

Запросы обычно рассматриваются как часть языка DML. Однако, так как запрос не меняет информацию в таблицах, а просто показывает ее пользователю, мы будем рассматривать запросы как самостоятельную категорию среди команд DML которые производят действие, а не просто показывают содержание базы данных.

Все запросы в SQL состоят из одиночной команды. Эта команда называется - SELECT (ВЫБОР).

Синтаксис команды:{IDISTINCT | ALL] < value expression>..}/\*

[INTO <host variable list> (\*embedded only\*)]< table reference>.

[WHERE <predicate>]

[GROUP BY <grouping column>..]

[HAVING <predicate>]

[ORDER BY <ordering column> [ASC | DESC]..];

Применяются следующие правила:

Если ни ALL, ни DISTINCT - не указаны, принимается - ALL.

Выражение <value expression> состоит из <column spec>, агрегатной функции<aggregate funct>, нестандартной функции <nonstandard function>, постоянной <constant>, или любой их комбинации с операторами в допустимых выражениях.

Ссылаемая таблица <table reference> состоит из имени таблицы, включая пре-фикс владельца, если текущий пользователь не владелец, или синоним (нестандартно) для таблицы. Таблица может быть или базовой таблицей или просмотром. В принципе, псевдоним может указать, какой синонимом используется для таблицы только на время текущей команды. Имя таблицы или синоним должны отделяться от псевдонима одним или более разделительными знаками <separator>.

Значения могут быть помещены и удалены из полей, тремя командами языка DML (Язык Манипулирования Данными):

INSERT (ВСТАВИТЬ),

UPDATE (МОДИФИЦИРОВАТЬ),

DELETE (УДАЛИТЬ).

INSERT (ВСТАВКА)

СинтаксисINTO <table name> (<column llst>) VALUES (<value llst>) | <query>;

INSERT создает одну или больше новых строк в таблице с именем <table name>.

Если используется предложение VALUES, их значения вставляются в таблицу с именем <table name>. Если запрос <query> указан, каждая строка вывода будет вставлена втаблицу с именем <table name>. Если список столбцов <column list> отсутствует, все столбцы таблицы <table name>, принимаются в упорядоченном виде.(МОДИФИКАЦИЯ)

Синтаксис<table name>

SET {<column name> = <value expression>}.

{[WHERE <predlcate>];} | {[WHERE CURRENT OF <cursorname>] <SQL term>]}

UPDATE изменяет значения в каждом столбце с именем <column name> на соответствующее значение <value expression>. Если предложение WHERE использует предикат <predicate>, то только строки таблиц чьи текущие значения делают тот предикат <predicate> верным, могут быть изменены. Если WHERE использует предложение CURRENT OF, то значения в строке таблицы с именем <table name> находящиеся в курсоре с именем <cursor name> меняются. WHERE CURRENT OF пригодно для использования только во вложенном SQL, и только с модифицируемыми курсорами.

При отсутствия предложения WHERE - все строки меняются.

**8. DDL**

запрос язык структурированный

В этом подразделении SQL рассматривается создание, изменение и удаление таблиц это относится к самим таблицам, а не к данным, которые в них содержатся.

Таблицы создаются командой CREATE TABLE. Эта команда создает пустую таблицу - таблицу без строк. Значения вводятся с помощью DML команды INSERT

Команда ALTER TABLE не часть стандарта ANSI; но это - широко доступная, и довольно содержательная форма, хотя ее возможности несколько ограничены. Она используется, чтобы изменить определение существующей таблицы. Обычно, она добавляет столбцы к таблице. Иногда она может удалять столбцы или изменять их размеры, а также в некоторых программах добавлять или удалять ограничения. Типичный синтаксис чтобы добавить столбец к таблице:

ALTER TABLE <table name> ADD <column name> <data type> <size>;

Синтаксис для удаления вашей таблицы, если конечно она является пустой, следующая:

DROP TABLE <table name>;

(УДАЛИТЬ)

СинтаксисFROM <table name> {[WHERE <predicate>];} | WHERE CURRENT OF

<cursor name><SQL term>

Если предложение WHERE отсутствует, ВСЕ строки таблицы удаляются. Если

предложение WHERE использует предикат <predicate>, строки, которые удовлетворяют условию этого предиката <predicate> удаляются. Если предложение WHERE имеет аргумент CURRENT OF (ТЕКУЩИЙ) в имени курсора <cursor name>, строка из таблицы <table name> на которую в данный момент имеется ссылка с помощью имени курсора <cursor name> будет удалена. Форма WHERE CURRENT может использоваться только во вложенном SQL, и только с модифицируемыми курсорами.

**9. DCD**

Операторы DCD, иногда называемые операторами Access Control Language, применяются для осуществления административных функций, присваивающих или отменяющих право (привилегию) использовать базу данных, таблицы в базе данных, а также выполнять те или иные операторы SQL.используется обычно в средах, которые требуют распознавания пользователей и различия между различными пользователями систем. Вообще говоря, администраторы баз данных, сами создают пользователей и дают им привилегии. С другой стороны пользователи которые создают таблицы, сами имеют права на управление этими таблицами. Привилегии - это то, что определяет, может ли указанный пользователь выполнить данную команду. Имеется несколько типов привилегий, соответствующих нескольким типам операций. Привилегии даются и отменяются двумя командами SQL: - GRANT (ДОПУСК) и REVOKE (ОТМЕНА).

Каждый пользователь в SQL базе данных имеет набор привилегий. Это-то, что пользователю разрешается делать (возможно это - файл регистрации, который может рассматриваться как минимальная привилегия). Эти привилегии могут изменяться со временем - новые добавляться, старые удаляться. Некоторые из этих привилегий определены в ANSI SQL, но имеются и дополнительные привилегии, которые являются также необходимыми.

Привилегии, которые можно назначить пользователю:Пользователь с этой привилегией может выполнять запросы в таблице.Пользователь с этой привилегией может выполнять команду INSERT

В таблице.Пользователь с этой привилегией может выполнять команду UPDATE на таблице. Вы можете ограничить эту привилегию для определенных столбцов таблицы.Пользователь с этой привилегией может выполнять команду DELETE в таблице.Пользователь с этой привилегией может определить внешний ключ, который использует один или более столбцов этой таблицы, как родительский ключ. Иногда, создателю таблицы хочется чтобы другие пользователи могли получить привилегии в его таблице. Обычно это делается в системах, где один или более людей создают несколько (или все) базовые таблицы в базе данных а затем передают ответственность за них тем кто будет фактически с ними работать. SQL позволяет делать это с помощью предложения WITH GRANT OPTION.

GRANT (ПЕРЕДАТЬ ПРАВА)

СинтаксисALL [PRIVILEGES] | {SELECT | INSERT | DELETE | UPDATE [(<columnllst>)] | REFERENCES [(<column llst>) l}.. ON <table name>.. TO| <Authorization ID>.. [WITH GRANT OPTION];

Аргумент ALL (ВСЕ), PRIVILEGES (ПРИВИЛЕГИИ), включает каждую привилегию в список привилегий. PUBLIC (ОБЩИЙ) включает всех существующих пользователей и всех созданных в будущем.

Эта команда дает возможность передать права для выполнения действий в таб-лице с указанным именем. REFERENCES позволяет дать права чтобы использовать столбцы в списке столбцов <column list> как родительский ключ для внешнего ключа.

Другие привилегии состоят из права выполнять команды, для которых привилегии указаны их именами в таблице. UPDATE подобен REFERENCES и может накладывать ограничения на определенные столбцы. GRANT OPTION дает возможность передавать эти привилегии другим пользователям.

Также как ANSI предоставляет команду CREATE TABLE чтобы создать таблицу, но не DROP TABLE чтобы от нее избавиться, так и команда GRANT позволяет вам давать привилегии пользователям, не предоставляя способа, чтобы отобрать их обратно. Потребность удалять привилегии сводится к команде REVOKE, фактически стандартному средству с достаточно понятной формой записи.

Синтаксис команды REVOKE - похож на GRANT, но имеет обратный смысл.(ОТМЕНИТЬ ПОЛНОМОЧИЯ)

Синтаксис

REVOKE {ALL [PRIVILEGES] | <privilege>..} [ON <table name>] FROM {| <Authorization ID>..};

Привилегия <privelege> может быть любой из указанных в команде GRANT. Пользователь, дающий REVOKE, должен иметь те же привилегии, что и пользователь, который давал GRANT. Предложение ON может быть использовано, если используется привилегия специального типа для особого объекта.

Другие операторы SQL

Кроме операторов рассмотренных выше и указанных в приложение А существует множество других, среди них:

средства управления транзакциями (см. таблицу 1)

средства администрирования данных (см. таблицу 2)

программный SQL (см. таблицу 3)

В коммерческих СУБД набор основных операторов расширен. В большинство СУБД включены операторы определения и запуска хранимых процедур и операторы определения тригеров.

Таблица 1 - средства управления транзакциями

|  |  |  |
| --- | --- | --- |
| **Оператор** | **Смысл** | **Действие** |
| COMMIT | Завершить транзакцию | Завершить комплексную взаимосвязанную обработку информации, объединенную в транзакцию |
| ROLLBACK | Откатить транзакцию | Отменить изменения, проведенные в ходе выполнения транзакции |
| SAVEPOINT | Сохранить промежуточную точку выполнения транзакции | Сохранить промежуточное состояние БД, пометить его для того, чтобы можно было в дальнейшем к нему вернутся |

Таблица 2 - средства администрирования данных

|  |  |  |
| --- | --- | --- |
| **Оператор** | **Смысл** | **Действие** |
| ALTER DATABASE | Изменить БД | Изменить набор основных объектов в базе данных, ограничении. |
| ALTER DBAREA | Изменить область хранения БД | Изменить ранее созданную область хранения |
| ALTER PASSWORD | Изменить пароль | Изменить пароль для всей базы данных |
| CREAT DBAREA | Создать БД | Создать новую базу данных, определив основные параметры для нее |
| CREAT DATABASE | Создать область хранения | Создать новую область хранения и сделать ее доступной для размещения данных |
| DROP DATABASE | Удалит БД | Удалить существующую базу данных ( |
| DROP DBAREA | Удалить область хранения БД | Удалить соответствующую область |

Таблица 3 - программный SQL

|  |  |  |
| --- | --- | --- |
| **Оператор** | **Смысл** | **Действие** |
| DECLARE | Определяет курсор для запроса | Задает некоторое имя и определяет связанный с ним запрос к БД, который соответствует виртуальному набору данных |
| OPEN | Открыть курсор | Формирует виртуальный набор данных, соответствующий описанию указанного курсора и текущего состоянию БД |
| FETCH | Считать строку из множества строк, определенных курсором | Считывает очередную строку, заданную параметром команды из виртуального набора данных, соответствующего открытому курсору |
| CLOSE | Закрыть курсор | Прекращает доступ к виртуальному набору данных, соответствующему указанному курсору |
| PREPARE | Подготовить оператор SQL к динамическому выполнению | Сгенерировать план выполнения запроса, соответствующего заданному оператору SQL |
| EXECUTE | Выполнить оператор SQL ранее подготовленный к динамическому выполнению | Выполняет ранее подготовленный план запроса |

**Заключение**

Возможно язык SQL не относится к семейству наиболее красивых, элегантных, понятных и приятных изобретений человечества. Тем не менее именно этот язык лежит в основе современных систем управления базами данных, и в ближайшем будущем эта ситуация сохранится.

Недаром теперь все чаще название языка понимается как Standard Query Language.

К достоинствам языка SQL можно отнести:

реляционный язык БД;

операторы манипулирования БД;

наличие международных стандартов SQL;

средства определения и манипулирования схемой БД;

определение ограничений целостности;

определение представлений;

определение индексов;

авторизация доступа к отношениям и их полям;

точки сохранения транзакций и откаты;

повсеместная распространенность;

быстрое обучение в простых случаях;

связывание с различными языками программирования;

поддержка ODBC и JDBC.

К недостаткам языка SQL можно отнести:

неполное соответствие реляционной модели данных (наличие дубликатов, необязательность первичного ключа, возможность упорядочения результатов);

недостаточно продуманный механизм неопределенных значений;

сложность формулировок и громоздкость.

Мир баз данных становится все более и более единым, что привело к необходимости создания стандартного языка, который мог бы использоваться, чтобы функционировать в большом количестве различных видов компьютерных сред.

Стандартный язык позволит пользователям знающим один набор команд, использовать их чтобы создавать, отыскивать, изменять, и передавать информацию независимо от того работают ли они на персональном компьютере, сетевой рабочей станции, или на универсальной ЭВМ. В нашем все более и более взаимосвязанном компьютерном мире, пользователь снабженный таким языком, имеет огромное преимущество в использовании и обобщении информации из ряда источников с помощью большого количества способов.

В компьютерном мире пользователь, владеющий таким языком, имеет огромные возможности по применению и интеграции информации из множества разнообразных источников.

Элегантность и независимость от специфики компьютерных технологий, а также его поддержка лидерами промышленности в области технологии реляционных баз данных, сделало SQL, и вероятно в течение обозримого будущего оставит его, основным стандартным языком. По этой причине, любой, кто хочет работать с базами данных в начале 21 века должен знать SQL.

**Список источников**

1. Глушаков С.В. Базы данных. Учебный курс [текст] Глушаков С.В., Ломотько Д.В.М.: Издательство «АСТ», 2000. - 504 с.
2. Харрингтон Джен Л. Проектирование реляционных баз данных: [текст] Лори, 2000.-230 стр.
3. Робинсон С. Microsoft Access 2000: учебный курс [текст] Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. - СПб.:Питер, 2002. - 512 с.
4. Хомоненко А.Д. Базы данных. [текст] - СПб.: КОРОНА принт, 2003.
5. Дейт К. Введение в системы баз данных, 6-е издание. [текст] СПб.: Издательский дом «Вильямс». 1999. - 848 с.
6. Грей Д. Управление данными - прошлое, настоящее и будущее [текст] Журнал «Системы управления базами данных», 1998 №3.
7. Груббер М. Понимание SQL [текст] M.:1993
8. Сайт http://www.compress.ru [Электронный ресурс]
9. Сайт http://www.MYSQL.ru [Электронный ресурс]
10. Сайт http://www.SQL.ru [Электронный ресурс]