ОСНОВЫ SQL

СПРАВКА

- iBase.ru
- Стандартные логин и пароль для IB: SYSDBA masterkey
- sql-school.info
- https://downloads.embarcadero.com/free/interbase
- *Бен Форта.* Освой самостоятельно язык запросов SQL / Пер. с англ. — 3-е изд. — 288 с.
- *Пол Уилтон, Джон Колби.* Язык запросов SQL для начинающих / Пер. с англ. — 496 с.
- *К. Дж. Дейт*. Введение в системы баз данных / Пер. с англ. — 1328 с.
- *Кевин Клайн. SQL. Справочник*. — 832 с.

СПРАВКА

- **Сайты СУБД**
- IBM DB2 - (http://ibm.com/db2)
- Microsoft Access - (http://office.microsoft.com/ru-ru/access/FX100487571049.aspx)
- Microsoft SQL Server - (http://www.microsoft.com/Rus/sql/)
- MySQL - (http://mysql.com/)
- Oracle - (http://www.oracle.com/global/ru/index.html)
- PostgreSQL - (http://www.postgresql.org/)
- Sybase Adaptive Server - (http://www.sybase.com/)

ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

- *Structured Query Language* — «язык структурированных запросов
- SQL является- информационно-логический язык, предназначенный для описания хранимых данных, для извлечения хранимых данных и для модификации данных.

Первый официальный стандарт языка SQL был принят ANSIПервый официальный стандарт языка SQL был принят ANSI(Американский Национальный Институт Стандартов) в 1986 году и *ISO* (Международной организацией по стандартизации) в 1987 году (так называемый *SQL-86*) и несколько уточнён в 1989 году) и несколько уточнён в 1989 году. Дальнейшее развитие языка поставщиками СУБД потребовало принятия в 1992 году нового расширенного стандарта (ANSI SQL-92) и несколько уточнён в 1989 году. Дальнейшее развитие языка поставщиками СУБД потребовало принятия в 1992 году нового расширенного стандарта (ANSI SQL-92 или просто SQL2). Следующим стандартом стал SQL:1999) и несколько уточнён в 1989 году. Дальнейшее развитие языка поставщиками СУБД потребовало принятия в 1992 году нового расширенного стандарта (ANSI SQL-92 или просто SQL2). Следующим стандартом стал SQL:1999 (SQL3). В настоящее время действует стандарт, принятый в 2003 году (SQL:2003) с небольшими модификациями, внесёнными позже.

ИСТОРИЯ ВЕРСИЙ СТАНДАРТА SQL

- 1986. SQL-86,SQL-87. Первый вариант стандарта, принятый институтом ANSI и одобренный ISO в 1987 году.
- 1989. SQL-89,FIPS 127-1. Немного доработанный вариант предыдущего стандарта.
- *1992*. SQL-92. SQL2, FIPS 127-2. Значительные изменения (ISO 9075); уровень *Entry Level* стандарта SQL-92 был принят как стандарт FIPS 127-2.
- *1999*. SQL:1999SQL:1999. SQL3. Добавлена поддержка регулярных выраженийSQL:1999. SQL3. Добавлена поддержка регулярных выражений, рекурсивныхSQL:1999. SQL3. Добавлена поддержка регулярных выражений, рекурсивных запросов, поддержка триггеровSQL:1999. SQL3. Добавлена поддержка регулярных выражений, рекурсивных запросов, поддержка триггеров, базовые процедурные расширения, нескалярные типы данных и некоторые объектно-ориентированные возможности.
- *2003*. SQL:2003SQL:2003. Введены расширения для работы с XMLSQL:2003. Введены расширения для работы с XML-данными, оконные функции (применяемые для работы с OLAP-базами данных), генераторы последовательностей и основанные на них типы данных.
- *2006*2006. *SQL:2006*. Функциональность работы с XML-данными значительно расширена. Появилась возможность совместно использовать в запросах SQL и XQuery.
- *2008*. SQL:2008. Улучшены возможности оконных функций, устранены некоторые неоднозначности стандарта SQL:2003

СООТВЕТСТВИЕ СТАНДАРТУ

Впервые понятие «уровня соответствия» было предложено в стандарте SQL-92. ANSI определяли четыре уровня соответствия реализации этому стандарту, где каждый последующий уровень соответствия заведомо подразумевал соответствие предыдущему уровню :

- *Entry* (базовый)
- *Transitional* (переходный) — проверку на соответствие этому уровню проводил только NIST
- *Intermediate* (промежуточный)
- *Full* (полный)

Положение изменилось с введением стандарта SQL:1999. в котором стандарт приобрёл модульную структуру — основная часть стандарта была вынесена в раздел «SQL/Foundation», все остальные были выведены в отдельные модули. Соответственно, остался только один уровень совместимости — что означает поддержку этой основной части. Поддержка остальных возможностей оставлена на усмотрение производителей СУБД. Аналогичное положение имело место и с последующими версиями стандарта.

ДОСТОИНСТВА SQL

- **Независимость от конкретной СУБД**
- **Наличие стандартов**
- **Декларативность**

НЕДОСТАТКИ SQL

- Несоответствие реляционной модели данных

Неопределённые значения (nulls)

Явное указание порядка колонок слева направо

Колонки без имени и дублирующиеся имена колонок

Отсутствие поддержки свойства «=»

Использование указателей

Высокая избыточность

- Сложность
- Отступления от стандартов
- Сложность работы с иерархическими структурами

РАСШИРЕНИЯ SQL

- InterBaseInterBase/Firebird – краткое название языка –*PSQL*. Procedural SQL
- IBM DB2 – краткое название языка – *SQL PL*. SQL Procedural Language (расширяет SQL/PSMSQL Procedural Language (расширяет SQL/PSM); также в DB2 хранимые процедуры могут писаться на обычных языках программирования: СиSQL Procedural Language (расширяет SQL/PSM); также в DB2 хранимые процедуры могут писаться на обычных языках программирования: Си, Java и т. д.
- MS SQL Server/ *Sybase ASE* – краткое название языка – Transact-SQL. Transact-SQL
- MySQLMySQL – краткое название языка – SQL/PSM. SQL/Persistent Stored Module
- OracleOracle – краткое название языка – PL/SQL. Procedural Language/SQL (основан на языке *Ada*)
- PostgreSQLPostgreSQL – краткое название языка – PL/pgSQL. Procedural Language/PostgreSQL Structured Query Language (очень похож на Oracle PL/SQL)

ОПЕРАТОРЫ SQL

Операторы SQL делятся на:

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

ОПЕРАТОРЫ *DDL*

- Create - СОЗДАТЬ
- Alter - ИЗМЕНИТЬ
- Drop - УДАЛИТЬ

ОБЪЕКТЫ БД

- DataBase – база данных БД
- Table - таблица
- Trigger - триггер
- Procedure - процедура
- Generator - генератор
- Exception – исключение
- VIEW - просмотры(вьюшки)

СОЗДАНИЕ ТАБЛИЦ

- CREATE TABLE имя_таблицы (
  имя_столбца тип_данных [NULL | NOT NULL] [CONSTRAINTS],
  имя_столбца тип_данных[NULL|NOT NULL] [CONSTRAINTS] , ...);
- Ограничения

На пустое значение

На значение по умолчанию

На определенные ранее параметры

ИЗМЕНЕНИЕ ТАБЛИЦ

- ALTER TABLE имя таблицы
  ADD | DROP имя_столбца тип_данных [NULL | NOT >NULL] [CONSTRAINTS] ,
  ADD I DROP имя_столбца тип_данных [NULL|NOT >NULL] [CONSTRAINTS] ,
- Изменения не только структуры но и целостности

УДАЛЕНИЕ ОБЪЕКТОВ

- DROP ТИП_ОБЪЕКТА имя_объекта;
- ТИП_ОБЪЕКТА:

Table - таблица

Trigger - триггер

Procedure - процедура

Generator - генератор

Exception – исключение

VIEW – вьюшка

INDEX – индекс

Foreign key – внешний ключ

Primary key – первичный ключ

СОЗДАНИЕ ОБЪЕКТОВ ТИПА PROCEDURE, TRIGGER

- CREATE ТИП_ОБЪЕКТА имя_объекта [параметры) [опции] AS SQL statement;
- Для Procedure
- Входные параметры
- Выходные параметры
- Принцип организационной структыр возвращаемых значений

СОЗДАНИЕ ТРИГГЕРОВ

- CREATE TRIGGER *name* FOR *table* [ACTIVE | INACTIVE] {BEFORE | AFTER} {DELETE | INSERT | UPDATE} [POSITION *number*] AS <*trigger_body*> *terminator* <*trigger_body*> = [<*variable_declaration_list*>] <*block*> <*variable_declaration_list*> = DECLARE VARIABLE *variable* <*datatype*>; [DECLARE VARIABLE *variable* <*datatype*>; ...] <*block*> = BEGIN <*compound_statement*> [<*compound_statement*> ...] END

СОЗДАНИЕ ГЕНЕРАТОРОВ

- CREATE GENERATOR имя_генератора;
- CREATE TRIGGER имя_триггера FOR имя_таблицы BEFORE INSERT POSITION 0

AS BEGIN

NEW. имя_поля = GEN_ID(имя_генератора, 1);

END;

ССЫЛОЧНАЯ ЦЕЛОСТНОСТЬ

- **CREATE TABLE ORDERS**
  (Order_ID integer,
  Order_Date date,
  Customer_SID integer,
  Amount double,
  Primary Key (Order_ID),
  Foreign Key (Customer_SID) references CUSTOMER(SID));

ПРИМЕР

- Table ***CUSTOMER***

SID Primary Key

Last_Name

First_Name

- Table ***ORDERS***

Order_ID Primary Key

Order_Date

Customer_SID Foreign Key

Amount

ВНЕШНИЙ КЛЮЧ

- **ALTER TABLE ORDERS**
  ADD FOREIGN KEY (customer_sid) REFERENCES CUSTOMER(SID);

ОПЕРАТОРЫ *DML*

- SELECT
- INSERT
- UPDATE
- DELETE

SELECT

- SELECT
  [ALL | DISTINCT | DISTINCTROW]
  [HIGH_PRIORITY]
  [STRAIGHT_JOIN]
  [SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
  [SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS] *выражение_Бе1ес1, ...* [INTO OUTFILE '*имя_файла*опции_экспорта*
  I INTO DUMPFILE ' *имя_файла} }*

[FROM *табличные_ссылки*
  [WHERE *определеяие_where]*
  [GROUP BY {имя_столбца *выражение позиция}* [ASC | DESCJ, ... [WITH ROLLUP]]
  [HAVING *onpsuejssHKS_wheze)*
  [ORDER BY {имя_столбца *выражение позиция]* [ASC | DESC] ,...]
  [LIMIT *[смещение,* {] количество_строк | количество_строк OFFSET *смещение}]*
  [PROCEDURE *имя_процедуры(список_аргументов)]*
  [FOR UPDATE | LOCK IN SHARE MODE]]

ПРИМЕР

- Firebird

SELECT **FIRST 10** * FROM [T]

- *Interbase*

SELECT * FROM [T] **ROWS 10**

- Microsoft

SELECT **TOP 10 [PERCENT]** * FROM T ORDER BY col

- MySQL

SELECT * FROM T **LIMIT 10**

- PostgreSQL

SELECT * FROM T **LIMIT 10**

- Oracle

SELECT * from T **WHERE ROWNUM <= 10**

INSERT

- INSERT [LOW_PRIORITY | DELAYED] [IGNORE] [INTO] tbl_name [(col_name,...)] VALUES (expression,...),(...),...
- INSERT [LOW_PRIORITY | DELAYED] [IGNORE] [INTO] tbl_name [(col_name,...)] SELECT ...
- INSERT [LOW_PRIORITY | DELAYED] [IGNORE] [INTO] tbl_name SET col_name=expression, col_name=expression, ...

РАСШИФРОВКА

- LOW_PRIORITY – низкий приоритет –будет выполнено когда нет клиентов
- DELAYED - позволяет пользователю продолжать работу сразу же не дожидаясь, пока инструкция вставки будет завершена
- IGNORE - строки значений, которых дублируют существующий ключ PRIMARY или UNIQUE в таблице, игнорируются и не будут вставлены, если не определяете IGNORE, вставка будет прервана, если имеется любая строка, которая дублирует существующее значение ключа.

UPDATE

- UPDATE [LOW_PRIORITY] [IGNORE] *шя_таблтш*
  SET *имя_столбца1=выражение1 [,*
  имя_*столбца2=выражение2 ...]*
  [WHERE *определение_where]*
  [ORDER BY ...]

- Многотабличный синтаксис:

UPDATE [LOW_PRIORITY] [IGNORE] имя_таблицы [,
имя_таблицы ...] SET *имя_столбца 1=выражение1 [,*
имя_столбца2=выражение2 ...] [WHERE определеяие_*where]*

DELETE

- DELETE [LOW_PRIORITY | QUICK] FROM table_name [WHERE where_definition] [ORDER BY ...]
- DELETE [LOW_PRIORITY | QUICK] table_name[.*] [table_name[.*] ...] FROM table-references [WHERE where_definition]

РАСШИФРОВКА

- QUICK - то драйвер таблицы не будет объединять индексные листья в течение процесса удаления, что может ускорять некоторые виды удаления