

Basic For Qt® and SQL

The following chapter has been used from Wikipedia entry about SQL and is licensed under the GNU Free Documentation License.

Contents

Definition.....	1
Data retrieval	1
Data manipulation	2
Data transaction	2
Data definition	3
Data control	3

Definition

SQL (commonly expanded to Structured Query Language) is the most popular computer language used to create, modify, retrieve and manipulate data from relational database management systems. The language has evolved beyond its original purpose to support object-relational database management systems. It is an ANSI/ISO standard.

SQL is designed for a specific, limited purpose — querying data contained in a relational database. As such, it is a set-based, declarative computer language rather than an imperative language such as C or BASIC which, being programming languages, are designed to solve a much broader set of problems. SQL keywords fall into several groups.

Data retrieval

The most frequently used operation in transactional databases is the data retrieval operation. When restricted to data retrieval commands, SQL acts as a functional language.

SELECT is used to retrieve zero or more rows from one or more tables in a database. In most applications, SELECT is the most commonly used DML command. In specifying a SELECT query, the user specifies a description of the desired result set, but they do not specify what physical operations must be executed to produce that result set. Translating the query into an efficient query plan is left to the database system, more specifically to the query optimizer.

Commonly available keywords related to SELECT include:

FROM is used to indicate from which tables the data is to be taken, as well as how the tables join to each other.

WHERE is used to identify which rows to be retrieved, or applied to GROUP BY. WHERE is evaluated before the GROUP BY.

GROUP BY is used to combine rows with related values into elements of a smaller set of rows.

HAVING is used to identify which of the “combined rows” (combined rows are produced when the query has a GROUP BY keyword or when the SELECT part contains aggregates), are to be retrieved. HAVING acts much like a WHERE, but it operates on the results of the GROUP BY and

hence can use aggregate functions.

ORDER BY is used to identify which columns are used to sort the resulting data.

Example 1:

```
SELECT * FROM books WHERE price > 100.00
ORDER BY title
```

This example shows the records for all books that cost more than 100.00 (units of currency). The result is sorted alphabetically by book title. The asterisk (*) means to show all columns of the referenced table(s). Alternatively, specific columns could be named. Example 2:

```
SELECT bk.title, count(*) AS Authors
FROM books AS bk, book_authors AS ba
WHERE bk.book_number = ba.book_number
GROUP BY bk.title
```

Example 2 shows both the use of multiple tables (called a “join”), and aggregation (grouping). “bk” and “ba” are known as “table aliases” and help avoid ambiguity when performing joins. (“AS” is optional in many dialects.) This example shows how many authors there are per book. Example output may resemble:

Title	Authors
SQL Examples and Guide	3
The Joy of SQL	1
How to use Wikipedia	2
Pitfalls of SQL	1
How SQL Saved my Dog	1

Data manipulation

First there are the standard Data Manipulation Language (DML) elements. DML is the subset of the language used to add, update and delete data.

INSERT is used to add zero or more rows (formally tuples) to an existing table.

UPDATE is used to modify the values of a set of existing table rows.

MERGE is used to combine the data of multiple tables. It is something of a combination of the INSERT and UPDATE elements. It is defined in the SQL:2003 standard; prior to that, some databases provided similar functionality via different syntax, sometimes called an “upsert”.

TRUNCATE deletes all data from a table (non-standard, but common SQL command).

DELETE removes zero or more existing rows from a table.

Example:

```
INSERT INTO my_table (field1, field2, field3) VALUES ('test', 'N', NULL);
UPDATE my_table SET field1 = 'updated value' WHERE field2 = 'N';
DELETE FROM my_table WHERE field2 = 'N';
```

Data transaction

Transaction, if available, can be used to wrap around the DML operations. START TRANSACTION (or BEGIN WORK, depending on SQL dialect) can be used to mark the start of a database transaction, which either completes completely or not at all.

COMMIT causes all data changes in a transaction to be made permanent.

ROLLBACK causes all data changes since the last COMMIT or ROLLBACK to be discarded, so

that the state of the data is “rolled back” to the way it was prior to those changes being requested. COMMIT and ROLLBACK interact with areas such as transaction control and locking. Strictly, both terminate any open transaction and release any locks held on data. In the absence of a START TRANSACTION or similar statement, the semantics of SQL are implementation-dependent.

Example:

```
START TRANSACTION;
UPDATE inventory SET quantity = quantity - 3 WHERE item = 'pants';
COMMIT;
```

Data definition

The second group of keywords is the Data Definition Language (DDL). DDL allows the user to define new tables and associated elements. Most commercial SQL databases have proprietary extensions in their DDL, which allow control over nonstandard features of the database system. The most basic items of DDL are the CREATE and DROP commands.

CREATE causes an object (a table, for example) to be created within the database.

DROP causes an existing object within the database to be deleted, usually irretrievably.

Some database systems also have an ALTER command, which permits the user to modify an existing object in various ways – for example, adding a column to an existing table. Example:

```
CREATE TABLE my_table (
my_field1    INT            UNSIGNED,
my_field2    VARCHAR (50),
my_field3    DATE           NOT NULL,
PRIMARY KEY (my_field1, my_field2)
)
```

Data control

The third group of SQL keywords is the Data Control Language (DCL). DCL handles the authorization aspects of data and permits the user to control who has access to see or manipulate data within the database. Its two main keywords are:

GRANT — authorises one or more users to perform an operation or a set of operations on an object.

REVOKE — removes or restricts the capability of a user to perform an operation or a set of operations.

Example:

```
GRANT SELECT, UPDATE ON my_table TO some_user, another_user
```