Learning SQL

Stephen CUI[1]

22 May, 2023

[1]cuixuanstephen@gmail.com

目录

Chapter 1

Creating and Populating a Database

1.1 Using the mysql Command-Line Tool

```
mysql -u root -p;
mysql -u root -p database_name;
```

1.2 MySQL Data Types

1.2.1 Character Data

Character data can be stored as either fixed-length or variable-length strings; the difference is that fixed-length strings are right-padded with spaces and always consume the same number of bytes, and variable-length strings are not right-padded with spaces and don't always consume the same number of bytes.

The maximum length for char columns is currently 255 bytes, whereas varchar columns can be up to 65,535 bytes. If you need to store longer strings (such as emails, XML documents, etc.), then you will want to use one of the text types (mediumtext and longtext).

> An exception is made in the use of varchar for Oracle Database. Oracle users should use the varchar2 type when defining variable-length character columns.

Character sets

You may choose to use a different character set for each character column in your database, and you can even store different character sets within the same table. To choose a character set other than the default when defining a column, simply name one of the supported character sets after the type definition.

```
varchar(20) character set latin1
```

With MySQL, you may also set the default character set for your entire database:

```
create database european_sales character set latin1;
```

Text data

If you need to store data that might exceed the 64 KB limit for varchar columns, you will need to use one of the text types.

When choosing to use one of the text types, you should be aware of the following:

表 1.1: MySQL text types

Text type	Maximum number of bytes
tinytext	255
text	65,535
mediumtext	16,777,215
longtext	4,294,967,295

- If the data being loaded into a text column exceeds the maximum size for that type, the data will be truncated.
- Trailing spaces will not be removed when data is loaded into the column.
- When using text columns for sorting or grouping, only the first 1,024 bytes are used, although this limit may be increased if necessary.

1.3 Table Creation

1.4 Populating and Modifying Tables

1.4.1 Inserting Data

There are three main components to an insert statement:

- The name of the table into which to add the data
- The names of the columns in the table to be populated
- The values with which to populate the columns

Generating numeric key data

If you are running these statements in your database, you will first need to disable the foreign key constraint on the table, and then re-enable the constraints when finished. The progression of statements would be:

```
set foreign_key_checks=0;
```

Chapter 2

Query Primer

2.1 Query Clauses

Several components or clauses make up the select statement. While only one of them is mandatory when using MySQL (the select clause), you will usually include at least two or three of the six available clauses. Table 2.1 shows the different clauses and their purposes.

2.2 The select Clause

The job of the select clause is as follows:

> The select clause determines which of all possible columns should be included in the query's result set.

If you were limited to including only columns from the table or tables named in the from clause, things would be rather dull. However, you can spice things up in your select clause by including things such as:

- Literals, such as numbers or strings
- Expressions, such as transaction.amount * -1
- Built-in function calls, such as ROUND(transaction.amount, 2)
- User-defined function calls

If you only need to execute a built-in function or evaluate a simple expression, you can skip the from clause entirely.

表 2.1: Query clauses Clause name Purpose

Clause name	Purpose
select	Determines which columns to include in the query's result set
from	Identifies the tables from which to retrieve data and how the tables should be joined
where	Filters out unwanted data
group by	Used to group rows together by common column values
having	Filters out unwanted groups
order by	Sorts the rows of the final result set by one or more columns

Removing Duplicates

> Keep in mind that generating a distinct set of results requires the data to be sorted, which can be time consuming for large result sets. Don't fall into the trap of using distinct just to be sure there are no duplicates; instead, take the time to understand the data you are working with so that you will know whether duplicates are possible.

2.3 The from Clause

2.3.1 Tables

Four different types of tables meet this relaxed definition:

- Permanent tables (i.e., created using the create table statement)
- Derived tables (i.e., rows returned by a subquery and held in memory)
- Temporary tables (i.e., volatile data held in memory)
- Virtual tables (i.e., created using the create view statement)

Temporary tables

These tables look just like permanent tables, but any data inserted into a temporary table will disappear at some point (generally at the end of a transaction or when your database session is closed).

Views

A view is a query that is stored in the data dictionary. It looks and acts like a table, but there is no data associated with a view (virtual table). When you issue a query against a view, your query is merged with the view definition to create a final query to be executed.

2.4 The where Clause

> The where clause is the mechanism for filtering out unwanted rows from your result set.

2.5 The order by Clause

> The order by clause is the mechanism for sorting your result set using either raw column data or expressions based on column data.

2.5.1 Sorting via Numeric Placeholders

If you are sorting using the columns in your select clause, you can opt to reference the columns by their position in the select clause rather than by name. This can be especially helpful if you are sorting on an expression.

Chapter 3

Filtering

3.1 Range Conditions

The between operator

When you have both an upper and lower limit for your range, you may choose to use a single condition that utilizes the between operator rather than using two separate conditions.

String ranges

To work with string ranges, you need to know the order of the characters within your character set (the order in which the characters within a character set are sorted is called a collation(排序规则)).

3.2 Membership Conditions

In some cases, you will not be restricting an expression to a single value or range of values but rather to a finite set of values.

Using subqueries

Along with writing your own set of expressions, such as ('G','PG'), you can use a subquery to generate a set for you on the fly.

Using not in

Sometimes you want to see whether a particular expression exists within a set of expressions, and sometimes you want to see whether the expression does not exist within the set.

3.3 Matching Conditions

Using wildcards

When building conditions that utilize search expressions, you use the like operator.

表 3.1: Wildcard characters

Wildcard character	Matches
_	Exactly one character
%	Any number of characters (including 0)

表 3.2: Sample search expressions

Search expression	Interpretation
F%	Strings beginning with F
%t	Strings ending with t
%bas%	Strings containing the substring 'bas'
__t_	Four-character strings with a t in the third position
___-__-____	11-character strings with dashes in the fourth and seventh positions

3.4 Null: That Four-Letter Word

null is a bit slippery, however, as there are various flavors of null:

Not applicable: Such as the employee ID column for a transaction that took place at an ATM machine

Value not yet known: Such as when the federal ID is not known at the time a customer row is created

Value undefined: Such as when an account is created for a product that has not yet been added to the database

When working with null, you should remember:

- An expression can be null, but it can never equal null.
- Two nulls are never equal to each other.

When working with a database that you are not familiar with, it is a good idea to find out which columns in a table allow nulls so that you can take appropriate measures with your filter conditions to keep data from slipping through the cracks.

Chapter 4

Querying Multiple Tables

4.1 Inner Joins

If you do not specify the type of join, then the server will do an inner join by default.

If the names of the columns used to join the two tables are identical, you can use the using subclause instead of the on subclause.

4.2 Joining Three or More Tables

Does Join Order Matter?

Keep in mind that SQL is a nonprocedural language, meaning that you describe what you want to retrieve and which database objects need to be involved, but it is up to the database server to determine how best to execute your query.

If, however, you believe that the tables in your query should always be joined in a particular order, you can place the tables in the desired order and then specify the keyword straight_join in MySQL, request the force order option in SQL Server, or use either the ordered or the leading optimizer hint in Oracle Database.

4.2.1 Using Subqueries as Tables

Chapter 5

Working with Sets

Although you can interact with the data in a database one row at a time, relational databases are really all about sets.

5.1 Set Theory Primer

The shaded area in Figure 5.1 represents the union of sets A and B, which is the combination of the two sets (with any overlapping regions included only once).

5.2 Set Theory in Practice

When performing set operations on two data sets, the following guidelines must apply:

- Both data sets must have the same number of columns.
- The data types of each column across the two data sets must be the same (or the server must be able to convert one to the other).

5.3 Set Operators

The SQL language includes three set operators that allow you to perform each of the various set operations described earlier in the chapter. Additionally, each set operator has two flavors, one that includes duplicates and another that removes duplicates (but not necessarily all of the duplicates).

5.3.1 The union Operator

The union and union all operators allow you to combine multiple data sets. The difference between the two is that union sorts the combined set and removes duplicates, whereas union all does not. With union all, the number of rows in the final data set will always equal the sum of the number of rows in the sets being combined.

If you would like your combined table to exclude duplicate rows, you need to use the union operator instead of union all.

5.3.2 The intersect Operator

The ANSI SQL specification includes the intersect operator for performing intersections. Unfortunately, version 8.0 of MySQL does not implement the intersect operator.

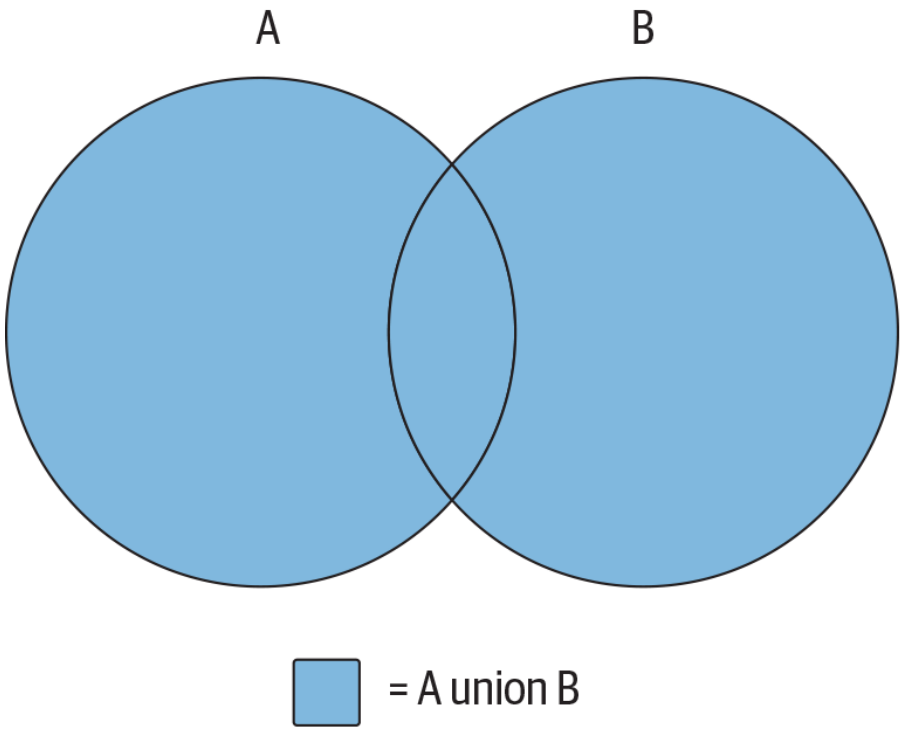

图 5.1: The union operation

Along with the intersect operator, which removes any duplicate rows found in the overlapping region, the ANSI SQL specification calls for an intersect all operator, which does not remove duplicates. The only database server that currently implements the intersect all operator is IBM’s DB2 Universal Server.

5.3.3 The except Operator

The ANSI SQL specification includes the except operator for performing the except operation. Once again, unfortunately, version 8.0 of MySQL does not implement the except operator.

There is also an except all operator specified in the ANSI SQL specification, but once again, only IBM’s DB2 Universal Server has implemented the except all operator.

The except all operator is a bit tricky, the difference between the two operations is that except removes all occurrences of duplicate data from set A, whereas except all removes only one occurrence of duplicate data from set A for every occurrence in set B.

5.4 Set Operation Rules

The following sections outline some rules that you must follow when working with compound queries.

5.4.1 Sorting Compound Query Results

If you want the results of your compound query to be sorted, you can add an order by clause after the last query. When specifying column names in the order by clause, you will need to choose from the column names in the first query of the compound query. Frequently, the column names are the same for both queries in a compound query, but this does not need to be the case.

5.4.2 Set Operation Precedence

If your compound query contains more than two queries using different set operators, you need to think about the order in which to place the queries in your compound statement to achieve the desired results.

In general, compound queries containing three or more queries are evaluated in order from top to bottom, but with the following caveats:

- The ANSI SQL specification calls for the intersect operator to have precedence over the other set operators.
- You may dictate the order in which queries are combined by enclosing multiple queries in parentheses.

MySQL does not yet allow parentheses in compound queries, but if you are using a different database server, you can wrap adjoining queries in parentheses to override the default top-to-bottom processing of compound queries.

Chapter 6

Data Generation, Manipulation, and Conversion

6.1 Working with String Data

When working with string data, you will be using one of the following character data types:

CHAR Holds fixed-length, blank-padded strings. MySQL allows CHAR values up to 255 characters in length, Oracle Database permits up to 2,000 characters, and SQL Server allows up to 8,000 characters.

varchar Holds variable-length strings. MySQL permits up to 65,535 characters in a var char column, Oracle Database (via the varchar2 type) allows up to 4,000 characters, and SQL Server allows up to 8,000 characters.

text (MySQL and SQL Server) or clob (Oracle Database) Holds very large variable-length strings (generally referred to as documents in this context). MySQL has multiple text types (tinytext, text, mediumtext, and longtext) for documents up to 4 GB in size. SQL Server has a single text type for documents up to 2 GB in size, and Oracle Database includes the clob data type, which can hold documents up to a whopping 128 TB. SQL Server 2005 also includes the varchar(max) data type and recommends its use instead of the text type, which will be removed from the server in some future release.

6.1.1 String Generation

The simplest way to populate a character column is to enclose a string in quotes.

When inserting string data into a table, remember that if the length of the string exceeds the maximum size for the character column (either the designated maximum or the maximum allowed for the data type), the server will throw an exception. Although this is the default behavior for all three servers, you can configure MySQL and SQL Server to silently truncate the string instead of throwing an exception.

Since MySQL 6.0, the default behavior is now “strict” mode, which means that exceptions are thrown when problems arise, whereas in older versions of the server the string would have been truncated and a warning issued. If you would rather have the engine truncate the string and issue a warning instead of raising an exception, you can opt to be in ANSI mode.

Including single quotes

Since strings are demarcated by single quotes, you will need to be alert for strings that include single quotes or apostrophes.

To make the server ignore the apostrophe in the word doesn't, you will need to add an escape to the string so that the server treats the apostrophe like any other character in the string. All three servers allow you to escape a single quote by adding another single quote directly before.

If you retrieve a string for use in a screen or report field, you don't need to do anything special to handle embedded quotes.

However, if you are retrieving the string to add to a file that another program will read, you may want to include the escape as part of the retrieved string. If you are using MySQL, you can use the built-in function quote(), which places quotes around the entire string and adds escapes to any single quotes/apostrophes within the string.

When retrieving data for data export, you may want to use the quote() function for all non-system-generated character columns, such as a customer_notes column.

Including special characters

When working with the French and German languages, for example, you might need to include accented characters such as é and ö. The SQL Server and MySQL servers include the built-in function char() so that you can build strings from any of the 255 characters in the ASCII character set (Oracle Database users can use the chr() function).

You can use the concat() function to concatenate individual strings, some of which you can type while others you can generate via the char() function.

> Oracle Database users can use the concatenation operator (||) instead of the concat() function.
> SQL Server does not include a concat() function, so you will need to use the concatenation operator (+),

If you have a character and need to find its ASCII equivalent, you can use the ascii() function, which takes the leftmost character in the string and returns a number.

6.1.2 String Manipulation

String functions that return numbers

Of the string functions that return numbers, one of the most commonly used is the length() function, which returns the number of characters in the string (SQL Server users will need to use the len() function).

The MySQL server removes trailing spaces from char data when it is retrieved, however, so you will see the same results from all string functions regardless of the type of column in which the strings are stored.

Along with finding the length of a string, you might want to find the location of a substring within a string. If the substring cannot be found, the position() function returns 0.

> For those of you who program in a language such as C or C++, where the first element of an array is at position 0, remember when working with databases that the first character in a string is at position 1. A return value of 0 from instr() indicates that the substring could not be found, not that the substring was

found at the first position in the string.

If you want to start your search at something other than the first character of your target string, you will need to use the locate() function, which is similar to the position() function except that it allows an optional third parameter, which is used to define the search's start position. The locate() function is also proprietary(专有的), whereas the position() function is part of the SQL:2003 standard.

Oracle Database does not include the position() or locate() function, but it does include the instr() function, which mimics the position() function when provided with two arguments and mimics the locate() function when provided with three arguments. SQL Server also doesn't include a position() or locate() function, but it does include the charindx() function, which also accepts either two or three arguments similar to Oracle's instr() function.

Another function that takes strings as arguments and returns numbers is the string comparison function strcmp(). strcmp(), which is implemented only by MySQL and has no analog in Oracle Database or SQL Server, takes two strings as arguments and returns one of the following:

- -1 if the first string comes before the second string in sort order
- 0 if the strings are identical
- 1 if the first string comes after the second string in sort order

Along with the strcmp() function, MySQL also allows you to use the like and regexp operators to compare strings in the select clause. Such comparisons will yield 1 (for true) or 0 (for false). Therefore, these operators allow you to build expressions that return a number.

String functions that return strings

Like all functions that return a string, you can use concat() to replace the data stored in a character column.

Another common use for the concat() function is to build a string from individual pieces of data. The concat() function can handle any expression that returns a string and will even convert numbers and dates to string format.

While concat() is useful for adding characters to the beginning or end of a string,you may also have a need to add or replace characters in the middle of a string. All three database servers provide functions for this purpose, but all of them are different.

MySQL includes the insert() function, which takes four arguments: the original string, the position at which to start, the number of characters to replace, and the replacement string. Depending on the value of the third argument, the function may be used to either insert or replace characters in a string. With a value of 0 for the third argument, the replacement string is inserted, and any trailing characters are pushed to the right.

Oracle Database does not provide a single function with the flexibility of MySQL's insert() function, but Oracle does provide the replace() function, which is useful for replacing one substring with another.

The replace() function will replace every instance of the search string with the replacement string, so you need to be careful that you don't end up with more replacements than you anticipated.

SQL Server also includes a replace() function with the same functionality as Oracle's, but SQL Server also includes a function called stuff() with similar functionality to MySQL's insert() function.

Along with inserting characters into a string, you may have a need to extract a substring from a string. For this purpose, all three servers include the substring() function (although Oracle Database's version is called substr()), which extracts a specified number of characters starting at a specified position.

6.2 Working with Numeric Data

The main concern when storing numeric data is that numbers might be rounded if they are larger than the specified size for a numeric column. For example, the number 9.96 will be rounded to 10.0 if stored in a column defined as float(3,1).

6.2.1 Performing Arithmetic Functions

The modulo operator, which calculates the remainder when one number is divided into another number, is implemented in MySQL and Oracle Database via the mod() function.

While the mod() function is typically used with integer arguments, with MySQL you can also use real numbers.

SQL Server does not have a mod() function. Instead, the operator % is used for finding remainders. The expression 10 % 4 will therefore yield the value 2.

Another numeric function that takes two numeric arguments is the pow() function (or power() if you are using Oracle Database or SQL Server), which returns one number raised to the power of a second number.

6.2.2 Controlling Number Precision

When working with floating-point numbers, you may not always want to interact with or display a number with its full precision. Four functions are useful when limiting the precision of floating-point numbers: ceil(), floor(), round(), and truncate(). All three servers include these functions, although Oracle Database includes trunc() instead of truncate(), and SQL Server includes ceiling() instead of ceil().

You can use the round() function to round up or down from the midpoint between two integers. Using round(), any number whose decimal portion is halfway or more between two integers will be rounded up, whereas the number will be rounded down if the decimal portion is anything less than halfway between the two integers.