# Learning SQL

Stephen CUI[1]

22 May, 2023

[1]cuixuanstephen@gmail.com

# 目录


**1 Creating and Populating a Database** **1**

1.1 Using the mysql Command-Line Tool . 1

1.2 MySQL Data Types . . . . . 1

1.2.1 Character Data . . . . . 1

1.3 Table Creation . . . . . 2

1.4 Populating and Modifying Tables . . . . 2

1.4.1 Inserting Data . . . . . 2

**2 Query Primer** **3**

2.1 Query Clauses . . . . . 3

2.2 The select Clause . . . . . 3

2.3 The from Clause . . . . . 4

2.3.1 Tables . . . . . 4

2.4 The where Clause . . . . . 4

2.5 The order by Clause . . . . . 4

2.5.1 Sorting via Numeric Placeholders . . . . . 4

**3 Filtering** **5**

3.1 Range Conditions . . . . . 5

3.2 Membership Conditions . . . . . 5

3.3 Matching Conditions . . . . . 5

3.4 Null: That Four-Letter Word . . . . . 6

**4 Querying Multiple Tables** **7**

4.1 Inner Joins . . . . . 7

4.2 Joining Three or More Tables . . . . . 7

4.2.1 Using Subqueries as Tables . . 7

**5 Working with Sets** **8**

5.1 Set Theory Primer . . . . . 8

5.2 Set Theory in Practice . . . . . 8

5.3 Set Operators . . . . . 8

5.3.1 The union Operator . . . . . 8

5.3.2 The intersect Operator . . . . . 8

5.3.3 The except Operator . . . . . 9

5.4 Set Operation Rules . . . . . 9

5.4.1 Sorting Compound Query Results 9

5.4.2 Set Operation Precedence . . . 10

# 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.

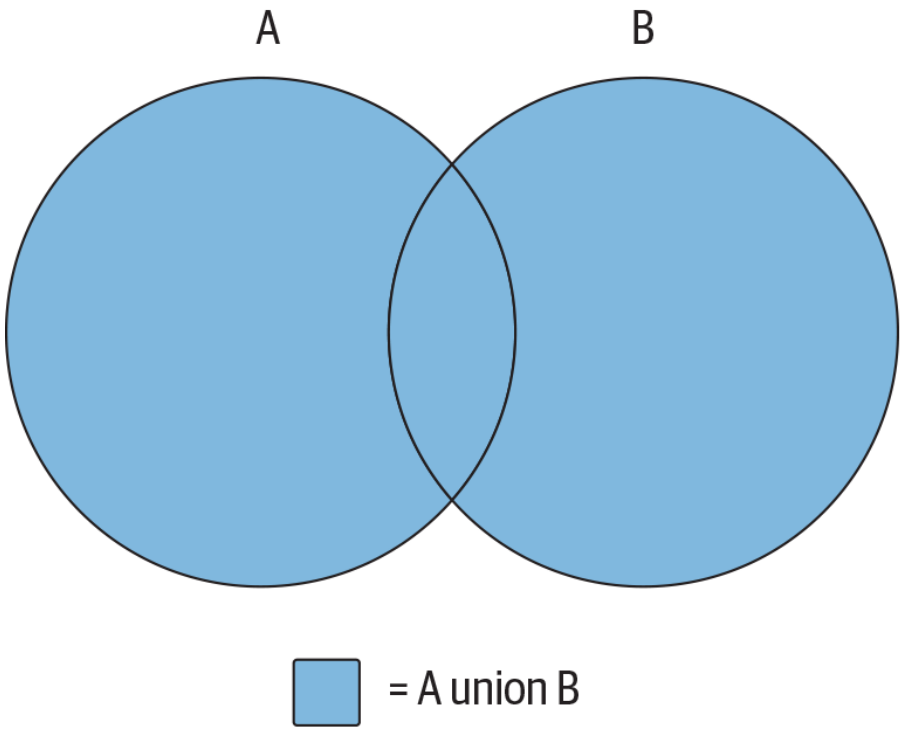


图 5.1: The union operation

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