**DS524 Data Management and Governance**

**HOS02A Introduction to MySQL Queries**

6/14/2021 Developed by Min Qiu

06/22/2021 Reviewed by Amanda Vaughan

01/10/2022 Reviewed by Mahathi Vucha

07/05/2023 Reviewed by Yared Shewarade

School of Technology & Computing @ City University of Seattle (CityU)



**Before You Start**

* The directory path shown in screenshots may be different from yours.
* Some steps are not explained in the tutorial**.** If you are not sure what to do:
  1. Consult the resources listed below.
  2. If you cannot solve the problem after a few tries, ask a TA for help.

**Learning Outcomes**

Students will be able to:

* Understand basic MySQL queries.
* Run queries in MySQL Workbench.
* Build a table.

**Resources**

* <https://www.sqltutorial.org/sql-select/>

**Introduction to SQL SELECT statement**

To query data from a table, you use the SQL SELECT statement. The SELECT statement contains the syntax for selecting columns, selecting rows, grouping data, joining tables, and performing simple calculations.

The SELECT statement is one of the most complex commands in SQL, therefore, in this tutorial, we will focus on the basics only.

The following illustrates the basic syntax of the SELECT statement that retrieves data from a single table:



In this syntax:

* First, specify a list of comma-separated columns from which you want to query the data in the SELECT clause.
* Then, specify the table name in the FROM clause.

When evaluating the SELECT statement, the database system evaluates the FROM clause first and then the SELECT clause.

The semicolon (;) is not part of the query. It is used to separate two SQL queries. Check out the [SQL syntax](https://www.sqltutorial.org/sql-syntax/) reference for more information.

In case you want to query data from all columns of a table, you can use the asterisk (\*) operator, like this:



Notice that SQL is case-insensitive. It means that the SELECT and select keywords are the same.

To make the SQL statements more readable, we will use the uppercase letters for the SQL keywords such as SELECT and FROM and the lowercase letters for the identifiers such as table and column names.

Besides the SELECT and FROM clauses, the SELECT statement can contain many other clauses such as:

* [WHERE](https://www.sqltutorial.org/sql-where/) – for filtering data based on a specified condition.
* [ORDER BY](https://www.sqltutorial.org/sql-order-by/) – for sorting the result set.
* [LIMIT](https://www.sqltutorial.org/sql-limit/) – for limiting rows returned.
* [JOIN](https://www.sqltutorial.org/sql-inner-join/)– for querying data from multiple related tables.
* [GROUP BY](https://www.sqltutorial.org/sql-group-by/) – for grouping data based on one or more columns.
* [HAVING](https://www.sqltutorial.org/sql-having/) – for filtering groups.

You will learn about the above clauses in the subsequent tutorials.

## SQL SELECT statement examples

We will use the sakila.actor table in the sample database for the demonstration purposes.

### SQL SELECT – querying data from all columns

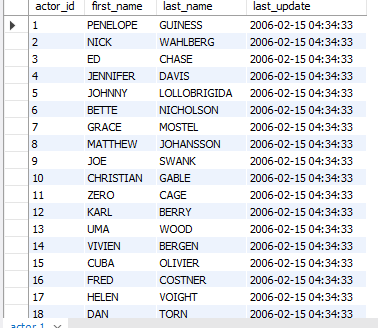
To query data from all columns of a table, you use an asterisk (\*) rather than listing all the column names.

The following example retrieves data from all the columns of the actor table:

Type the following code:

SELECT \* FROM actor;

The result set contains the data of the columns in the order which they were defined when the actor table was created:



Using the asterisk (\*) operator is only convenient for querying data interactively through an SQL client application.

However, if you use the asterisk (\*) operator in the embedded SQL statements in your application, you may have some potential problems.

The reason is that the table structure will evolve to adapt to the new business requirements e.g., you may add a new column or remove an existing column. If you use the asterisk (\*) and do not change the application code to make it work with the new table structure, the application may not work properly.

On top of this, using the asterisk (\*) might cause a performance issue. The application often does not need all data from all columns of a table. If you use the asterisk (\*), the database server must read the unnecessary data and this unnecessary data must transfer between the server and application. It causes slowness in the application.

### SQL SELECT – querying data from specific columns

The SELECT statement allows you to specify exactly which columns you want to retrieve data in any order. It does not have to be in the order defined in the table.

For example, if you use want to view the first name and last name of all actors, you can use the following query:

Type the following code:

SELECT

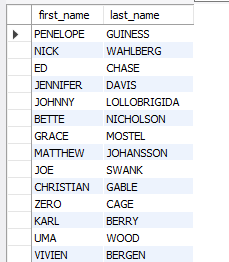
first\_name,

last\_name

FROM

actor;

Notice that the result set only includes the columns specified in the SELECT clause.



### SQL SELECT – querying data from specific scope

The WHERE statement allows you to specify conditions on the fields of your data.

For example, if you use want to view the amounts greater than 6, you can use the following query:

Type the following code:

SELECT

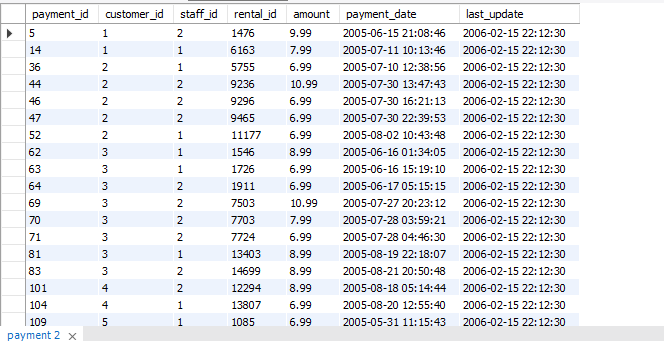
\*

FROM

Sakila.payment

WHERE

amount > 6;



If we want to look for a payment\_date that occurs on 2005-06-15:

Type the following code:

SELECT

\*

FROM

Sakila.payment

WHERE

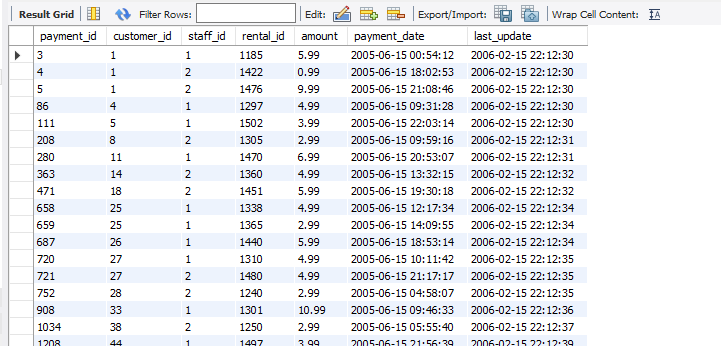
payment\_date

between

‘2005-06-15’

and

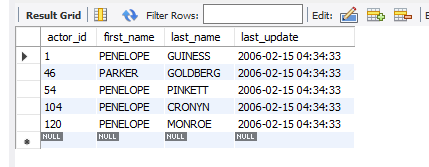
‘2005-06-16’;



**Using “Like” to find the keyword you want.**

Type the following code:

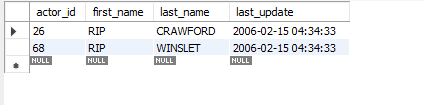
1) SELECT \* FROM sakila.actor where first\_name like “P%”;



This shows every first name that starts with “P”.

Type the following code:

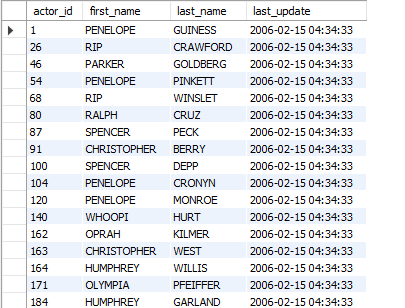
2) SELECT \* FROM sakila.actor where first\_name like “%P”;



This shows every first name that ends with “P”.

Type the following code:

3) SELECT \* FROM sakila.actor where first\_name like “%P%”;



This shows every first name that contains “P”.

You can try this command as well:

SELECT first\_name, last\_name FROM sakila.actor where first\_name like “%P%”;

**Submit your work to GitHub**

Follow the instructions on the CityU STC TA Center Github.io [Submit your work page](https://cityuseattle.github.io/docs/hoporhos/submit/).