MySQL Views & the WITH CHECK OPTION Clause

Introduction to MySQL View & the WITH CHECK OPTION clause

Sometimes, you [create a view](https://www.mysqltutorial.org/create-sql-views-mysql.aspx) to reveal the partial data of a table. However, a simple view is [updatable](https://www.mysqltutorial.org/create-sql-updatable-views.aspx) therefore it is possible to [update](https://www.mysqltutorial.org/mysql-update-data.aspx) data which is not visible through the view. This update makes the view inconsistent. To ensure the consistency of the view, you use the WITH CHECK OPTION clause when you [create or modify the view](https://www.mysqltutorial.org/managing-sql-views.aspx).

The WITH CHECK OPTION is an optional clause of the CREATE VIEW statement. The WITH CHECK OPTION prevents a view from [updating](https://www.mysqltutorial.org/mysql-update-data.aspx) or [inserting](https://www.mysqltutorial.org/mysql-insert-statement.aspx) rows that are not visible through it. In other words, whenever you update or insert a row of the base tables through a view, MySQL ensures that the insert or update operation is conformed with the definition of the view.

The following illustrates the syntax of the WITH CHECK OPTION clause.

**CREATE** [**OR** **REPLACE** **VIEW**] view\_name

**AS**

select\_statement

**WITH** **CHECK** **OPTION**;

Code language: SQL (Structured Query Language) (sql)

Notice that you put the semicolon (;) at the end of the WITH CHECK OPTION clause, not at the end of the [SELECT](https://www.mysqltutorial.org/mysql-select-statement-query-data.aspx) statement that defines the view.

Let’s take a look at an example of using the WITH CHECK OPTION clause.

MySQL view and WITH CHECK OPTION example

First, [create a view](https://www.mysqltutorial.org/create-sql-views-mysql.aspx) named vps based on the employees table to reveal employees whose job titles are VP e.g., VP Sales, VP Marketing.

**CREATE** **OR** **REPLACE** **VIEW** vps **AS**

**SELECT**

employeeNumber,

lastname,

firstname,

jobtitle,

extension,

email,

officeCode,

reportsTo

**FROM**

employees

**WHERE**

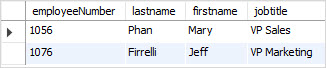
jobTitle **LIKE** '%VP%';

Code language: SQL (Structured Query Language) (sql)

Next, query data from the vps view using the following SELECT statement:

**SELECT** \* **FROM** vps;

Code language: SQL (Structured Query Language) (sql)



Because the vps is a simple view, it is updatable.

Then, [insert a row](https://www.mysqltutorial.org/mysql-insert-statement.aspx) into the employees table through the vps view.

**INSERT** **INTO** vps(

employeeNumber,

firstName,

lastName,

jobTitle,

extension,

email,

officeCode,

reportsTo

)

**VALUES**(

1703,

'Lily',

'Bush',

'IT Manager',

'x9111',

'lilybush@classicmodelcars.com',

1,

1002

);

Code language: SQL (Structured Query Language) (sql)

Notice that the newly created employee is not visible through the vps view because her job title is IT Manager, which is not the VP. You can verify it using the following SELECT statement.

**SELECT**

\*

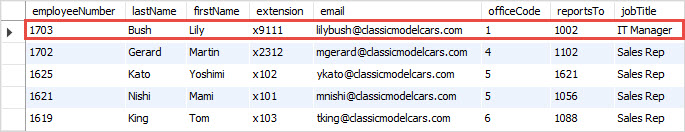
**FROM**

employees

**ORDER** **BY**

employeeNumber **DESC**;

Code language: SQL (Structured Query Language) (sql)



This may not what we want because we just want to expose the VP employees only through the vps view, not other employees.

To ensure the consistency of a view so that users can only display or update data that visible through the view, you use the WITH CHECK OPTION when you create or modify the view.

Let’s modify the view to include the WITH CHECK OPTION.

**CREATE** **OR** **REPLACE** **VIEW** vps **AS**

**SELECT**

employeeNumber,

lastName,

firstName,

jobTitle,

extension,

email,

officeCode,

reportsTo

**FROM**

employees

**WHERE**

jobTitle **LIKE** '%VP%'

**WITH** **CHECK** **OPTION**;

Code language: SQL (Structured Query Language) (sql)

Notice the WITH CHECK OPTION at the end of the CREATE OR REPLACE statement.

After that, [insert a row into](https://www.mysqltutorial.org/mysql-insert-statement.aspx) the employees table through the vps view:

**INSERT** **INTO** vps(employeeNumber,firstname,lastname,jobtitle,extension,email,officeCode,reportsTo)

**VALUES**(1704,'John','Smith','IT Staff','x9112','johnsmith@classicmodelcars.com',1,1703);

Code language: SQL (Structured Query Language) (sql)

This time, MySQL rejected the insert and issued the following error message:

Error Code: 1369. **CHECK** **OPTION** **failed** 'classicmodels.vps'

Code language: SQL (Structured Query Language) (sql)

Finally, insert an employee whose job title is SVP Marketing into the employees table through the vps view to see if it is allowed.

**INSERT** **INTO** vps(employeeNumber,firstname,lastname,jobtitle,extension,email,officeCode,reportsTo)

**VALUES**(1704,'John','Smith','SVP Marketing','x9112','johnsmith@classicmodelcars.com',1,1076);

Code language: SQL (Structured Query Language) (sql)

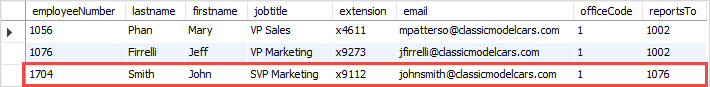
MySQL issued the message:

1 rows(s) affected.

You can verify the insert by querying data from the vps view.

**SELECT** \* **FROM** vps;

Code language: SQL (Structured Query Language) (sql)



It works as expected.