

Benefits of Using Views

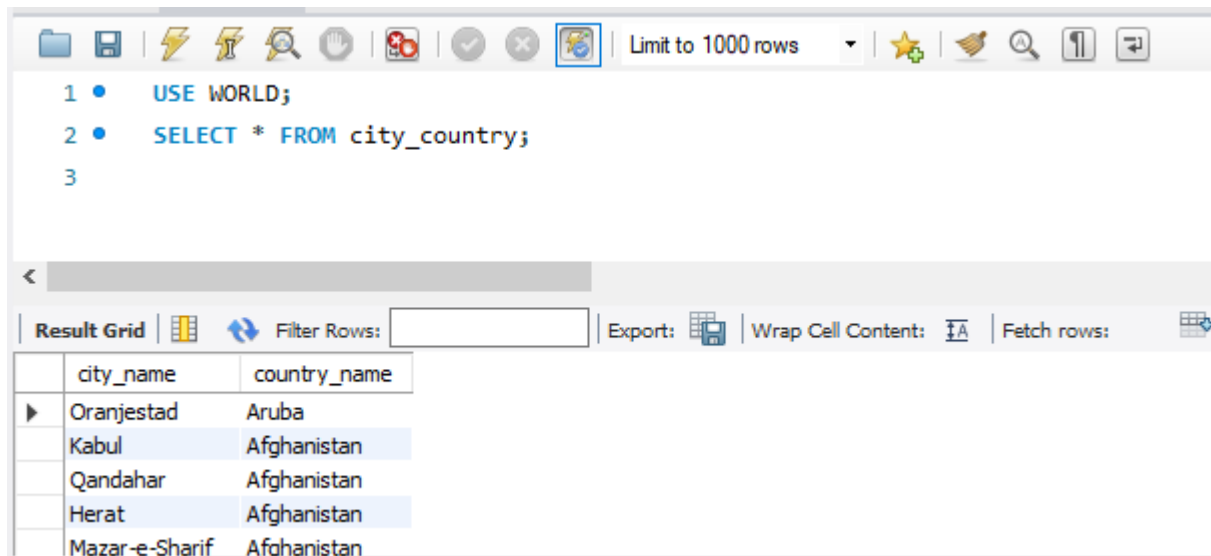
Benefits of Using Views

- **Design Flexibility:** By using a view instead of a query in an application, it is easier to make changes to the underlying table structure.
- **Improved Security:** By using a view to return data from tables instead of a SELECT, you can hide the WHERE clause or other columns to which you do not want the user to have access.
- **Query Simplification:** You can write simple select statements against views, which handle complex queries and joins.

Code Sample:

```
USE WORLD;  
CREATE VIEW city_country AS  
SELECT ci.name AS city_name, co.name AS country_name  
FROM city ci  
      JOIN country co  
      ON ci.CountryCode = co.Code;
```

Results by selecting from the city_country view:



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The query editor contains the following SQL statements:

```

1 • USE WORLD;
2 • SELECT * FROM city_country;
3

```

Below the query editor is a 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, a 'Wrap Cell Content' checkbox, and a 'Fetch rows' button. The result grid displays the following data:

| city_name | country_name |
|----------------|--------------|
| Oranjestad | Aruba |
| Kabul | Afghanistan |
| Qandahar | Afghanistan |
| Herat | Afghanistan |
| Mazar-e-Sharif | Afghanistan |

CREATE VIEW city_country AS

- Create a new VIEW object and give it the name city_country
- The AS statement precedes the query that will be assigned to the VIEW

SELECT ci.name AS city_name, co.name AS country_name

- Only the columns defined in the SELECT statement will be available to the VIEW
- It is a good idea to provide a column alias in the select because the VIEW will not have access to the underlying table structure.

FROM city ci

JOIN country co

ON ci.CountryCode = co.Code;

- The JOIN statement of the SELECT.
- Once you have created a VIEW, you can run SQL statements using the VIEW as if it were a table.
- By creating a VIEW, we can run selects that retrieve data from multiple tables without having to re-code a join.
- Notice how the SELECT * retrieves only the rows defined in the SELECT statement used in the VIEW creation.
- If you want to drop a VIEW, we can run the DROP VIEW statement
- If you want to modify an existing view you can use the statement CREATE OR REPLACE VIEW. That way you do not have to run a DROP VIEW statement and then a CREATE VIEW statement.





This content is provided to you freely by BYU-I Books.

Access it online or download it at https://books.byui.edu/learning_mysql/benefits_of_using_vi.

