Hello. In this tutorial, we will learn the <code>INTERSECT</code> operator in sql and will use the postgresql database running on Docker.

<h2>1. Introduction</h2>

<strong>SQL</strong> stands for <em>Structured Query Language</em> and is used to extract and organize data stored in relational databases like MySQL, PostgreSQL, Oracle, etc. A relational database consists of rows and columns that allow fetching specific information from databases that can be used later for analysis. In real-time SQL manages a large amount of data that is written and read simultaneously and any query that reaches the SQL server is processed into three parts –

<ul>

 <li><em>Parsing</em> – Process to check the SQL query syntax</li>

 <li><em>Binding</em> – Process to check the SQL query semantics</li>

 <li><em>Optimization</em> – Process to generate the SQL query execution plan</li>

</ul>

<h3>1.1 Usage of SQL</h3>

Structured Query Language (popularly known as <em>SQL</em>) is commonly used by data analysts and data science professionals and is helpful too –

<ul>

 <li>Execute queries against the database</li>

 <li>Retrieve data from the database</li>

 <li>Insert new records into the database</li>

 <li>Update existing records in the database</li>

 <li>Created stored procedures, functions, and materialized views in the database</li>

 <li>Create users and grant permissions</li>

 <li>Set permissions on tables, stored procedures, functions, and materialized views</li>

</ul>

<h3>1.2 INTERSECT operator in SQL</h3>

<ul>

 <li>The <code>INTERSECT</code> operator in sql combines two <code>SELECT</code> statements and returns only the dataset common in both statements</li>

 <li>Follow the two rules –

 <ul>

 <li>The number and order of columns in both queries have to be the same</li>

 <li>Data types of corresponding columns from both the <code>SELECT</code> statements must be compatible with each other</li>

 </ul>

</li>

</ul>

The <code>INTERSECT</code> operator is represented by the below syntax.

<span style="text-decoration: underline;"><em>Query Syntax</em></span>

<pre class="brush:sql; wrap-lines:false;">

SELECT column1 [, column2 ]

 FROM table1 [, table2 ]

 [WHERE condition]

 INTERSECT

 SELECT column1 [, column2 ]

 FROM table1 [, table2 ]

 [WHERE condition];

</pre>

<h2>2. SQL INTEREST operator</h2>

Let us dive into some practice implementation on the postgresql database.

<h3>2.1 Pre-requirement - Postgres Setup</h3>

Usually, setting up the database is a tedious step but with <a href="https://www.docker.com/" target="\_blank">Docker</a>, it is a simple process. You can watch the video available at this <a href="<https://www.youtube.com/watch?v=S7NVloq0EBc>" target="\_blank">link</a> to understand the Docker installation on Windows OS. Once done open the terminal and trigger the below command to set and run postgresql.

<span style="text-decoration: underline;"><em>Docker commands</em></span>

<pre class="brush:plain; wrap-lines:false;">

-- command to run postgres on docker –

-- remember to change the password --

docker run -d -p 5433:5432 -e POSTGRES\_PASSWORD=<your\_password> --name postgres postgres

-- command to stop the Postgres docker container --

docker stop postgres

-- command to remove the Postgres docker container --

docker rm postgres

</pre>

Remember to enter the password of your choice. If everything goes well the postgresql database server would be up and running on a port number – <code>5433</code> and you can connect with the <a href="https://dbeaver.io/" target="\_blank">Dbeaver</a> GUI tool for connecting to the server.

// Fig. 1. Postgres on Docker

<h3>2.2 Setting up pre-requisite data</h3>

Create the database named <code>learning</code> and add the following table and data to the tables with the help of the below sql script.

<span style="text-decoration: underline;"><em>Pre-requisite sql script</em></span>

<pre class="brush:sql; wrap-lines:false;">

-- customers

create table customers(id serial primary key, name varchar, country varchar, city varchar);

insert into customers (name, country, city)

values

('raltofts0', 'Thailand', 'Nang Rong'),('fgariff1', 'China', 'Yuefeng'),('csilcock2', 'France', 'Lacroix-Saint-Ouen'),('mhasluck3', 'Indonesia', 'Cibaregbeg Dua'),('bmerrall4', 'Russia', 'Ust’-Uda'),('scuddon5', 'Greece', 'Nestório'),('amichelet6', 'Vietnam', 'Quán Hành'),('hpoon7', 'Colombia', 'Sevilla'),('rchugg8', 'Madagascar', 'Sakaraha'),('pleyland9', 'Seychelles', 'Anse Boileau');

select id, name, country, city from customers;

-- branches

create table branches(id serial primary key, country varchar, city varchar);

insert into branches (country, city)

values

('Russia', 'Neftegorsk'),('Indonesia', 'Krajan Dua Dawuan Wetan'),('Russia', 'Perm'),('China', 'Daxing'),('China', 'Bianzhuang'),('Argentina', 'Apóstoles'),('China', 'Jianrao'),('Yemen', 'Al Jabīn'),('Finland', 'Muhos'),('Philippines', 'Kaytitinga'),('China', 'Tekes'),('Indonesia', 'Nanger'),('Russia', 'Podol’sk'),('Russia', 'Lensk'),('Finland', 'Orimattila'),('Vietnam', 'Phước An'),('China', 'Dadamtu'),('Poland', 'Potęgowo'),('Thailand', 'Fao Rai'),('Pakistan', 'Jhumra');

select id, country, city from branches;

</pre>

Once the sql script is executed the mock data will be successfully added to the <code>customers</code> and <code>branches</code> tables respectively.

<h3>2.3 Playing with the INTERSECT operator</h3>

Use the below <code>SELECT</code> keyword to understand the <code>INTERSECT</code> sql operator.

<span style="text-decoration: underline;"><em>UNION operator</em></span>

<pre class="brush:sql; wrap-lines:false;">

-- sql intersect

-- select query will display the common country records from both tables

select country

from customers

intersect select country

from branches

order by country;

-- select query will display the common country and city records from both tables

select country, city

from customers

intersect select country, city

from branches

order by city;

</pre>

If everything goes well the result of the sql query will be shown on the console and the usage of the <code>INTERSECT</code> operator can be practically understood.

// Fig. 2: INTERSECT operator

<h2>3. Summary</h2>

In this tutorial, we learned the brief of sql and sql <code>INTERSECT</code> operator via different examples. You can download the sql scripts used in this tutorial from the <a href="#projectDownload">Downloads</a> section.

<h2><a name="projectDownload"></a>4. Download the Sql scripts</h2>

This was a tutorial to understand the <code>INTERSECT</code> operator in sql.

<div class="download"><strong>Download</strong><br />You can download the full source code of this example here: </div>

----------------------------------------------------------------------------------------------------------------------------------

Download link name – <strong>SQL INTERSECT operator</strong>